Хмельницький національний університет
Факультет інформаційних технологій
Кафедра кібербезпеки

КВАЛІФІКАЦІЙНА РОБОТА

Кукурудзи Дмитра Миколайовича

на здобуття ступеня вищої освіти Бакалавра

Система моніторингу безпеки мережевої інфраструктури

Галузь знань 12 – Інформаційні технології

Спеціальність 125 –Кібербезпека

Освітня програма Кібербезпека

КРБКБ. 2102152.21.02.15 ПЗ

Виконав студент 4 курсу, група КБ-21-2 _____ Дмитро КУКУРУДЗА
Підпис, дата Ініціали, прізвище

Керівник к.т.н, доцент _____ Віра ТІТОВА
Науковий ступінь, вчене звання Підпис, дата Ініціали, прізвище

Нормоконтролер старший викладач _____ Сергій МОСТОВИЙ
Науковий ступінь, вчене звання Підпис, дата Ініціали, прізвище

До захисту допускаю:

Зав. кафедри кібербезпеки _____ Юрій КЛЬОЦ
Підпис, дата Ініціали, прізвище

16 06 2025р.

Хмельницький, 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій.

Кафедра Кібербезпеки.

Рівень вищої освіти Бакалавр.

Галузь знань 12 – Інформаційні технології.

Спеціальність 125 – Кібербезпека.

Освітня програма Кібербезпека.

ЗАТВЕРДЖУЮ

Зав. кафедри Ю.П. Кльоц

" 15 " 03 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Кукурудзі Дмитру Миколайовичу

1. Тема проекту (роботи) Система моніторингу безпеки мережевої інфраструктури.
Керівник проекту (роботи) Тітова Віра Юріївна к.т.н., доцент
Затверджена наказом ректора університету від 07.02.2025 р. № 23
2. Строк подання студентом проекту (роботи) на кафедру 2.06.2025 р.
3. Вихідні дані до проекту (роботи) Завдання кваліфікаційної роботи полягає в виборі системи моніторингу мережі та базової операційної системи, встановленні операційної системи та розгортання вибраної системи моніторингу мережі. Проведення базового конфігурування системи моніторингу мережі. Конфігурування мережевого обладнання, налаштування протоколів обміну даними та логування. Налаштування прийому логів та повідомлень за SNMP протоколом для опитування обладнання. Налаштування тригерів на виявлення в мережі сплесків активності, атипових дій та втрати зв'язку з проміжним обладнанням мережі. Налаштування оповіщення адміністраторів про виявлення аномалій в мережі та сценаріїв реакції на таку поведінку.
4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)
Вступ. Огляд відомих рішень систем моніторингу мережі. Обґрунтування вибору системи моніторингу мережі. Розгортання і базове налаштування системи моніторингу мережі. Налаштування опитувань та сповіщень, конфігурування мережевого обладнання. Тестування розгорнутої системи. Висновки.
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)
Структура системи моніторингу мережі. Фрагмент аналізованої мережі. Алгоритм роботи.

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль			
Антиплагіат			

7. Дата видачі завдання «_16_»_____лютого_____2025 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи з керівником	Лютий	
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	Березень	
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	Березень	
4	Робота над розділом 2 – формування вимог	Квітень	
5	Робота над розділом 3 –реалізація комп'ютерної мережі	Квітень	
6	Оформлення пояснювальної записки згідно вимог	Травень	
7	Попередній захист ВКР	Травень	
8	Захист ВКР на засіданні ЕК	Червень	

Студент _____ Підпис Д.М. Кукурудза Ініціали, прізвище

Керівник проекту (роботи) _____ Підпис В.Ю. Тітова Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Система моніторингу безпеки мережевої інфраструктури».

Автор роботи: студент групи КБ-21-2 Дмитро КУКУРУДЗА.

Керівник роботи: канд. тех. наук, доцент кафедри КБ Віра Юріївна Тітова.

Пояснювальна записка: *78 с.*, 18 *рис.*, 6 *табл.*, 3 *дод.*, 58 *джерел.*

Графічна частина: 3 креслення.

Моніторинг, мережа, безпека мережі, система моніторингу, LibreNMS, протокол SNMP.

Метою кваліфікаційної роботи є розробка та впровадження системи моніторингу безпеки мережевої інфраструктури з використанням відкритого програмного забезпечення LibreNMS. У роботі розглянуто архітектуру системи, можливості виявлення інцидентів, методи інтеграції з протоколами SNMP, Syslog, ICMP, а також реалізовано практичну частину з налаштуванням сповіщень, логуванням подій і побудовою візуалізацій.

Актуальність роботи зумовлена зростаючими вимогами до надійності й захищеності комп'ютерних мереж в умовах постійного збільшення кількості загроз. Застосування LibreNMS дозволяє створити адаптивну, масштабовану й технічно ефективну систему, що забезпечує постійний контроль за станом мережевого середовища, виявлення відхилень і реагування на інциденти безпеки в режимі реального часу. Розроблене рішення може бути використане в організаціях різного масштабу для підвищення стійкості IT-інфраструктури та зниження ризиків інформаційних втрат.

16.06.2025

ABSTRACT

Subject of qualification work: Network Infrastructure Security Monitoring System

Author: Dmytro KUKURUDZA

Head of work: Ph. D. tech. Sciences. Associate Professor of the Department of Cybersecurity: Vira Yuriyivna Titova

Explanatory note: 78 pages, 18 figures, 6 tables, 3 appendices, 58 references

Graphic section: 3 drawings

Key words: Monitoring, network, network security, monitoring system, LibreNMS, SNMP protocol

The purpose of this Bachelor's qualification workis the development and implementation of a network infrastructure security monitoring system using open-source software LibreNMS. The work explores the system architecture, incident detection capabilities, methods of integration with SNMP, Syslog, and ICMP protocols, as well as practical implementation of alerting, event logging, and data visualization.

The relevance of the research is determined by the growing demands for reliability and protection of computer networks in the face of continuously increasing threats. The use of LibreNMS enables the creation of an adaptive, scalable, and technically efficient system that ensures continuous control over the state of the network environment, detection of anomalies, and real-time response to security incidents. The developed solution can be applied in organizations of various scales to improve IT infrastructure resilience and reduce the risk of information loss.

16.06.2025

ЗМІСТ

					КРБКБ. 2101006.21.01.06 ПЗ		
Зм.	Арк	№докум.	Підпис	Дата			
Виконав		Кукурудза Д.М.		16.06.25	Система моніторингу безпеки мережевої інфраструктури Пояснювальна записка	Літера	Аркуш 6 / Аркушів 78
Перевір.		Тітова В.Ю.		16.06.25			
Н.контр.		Мостовий С.В.		16.06.25		ХНУ, КБ-21-2	
Затвер.		Кльоц Ю.П.		16.06.25			

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

SNMP – Simple Network Management Protocol

ICMP – Internet Control Message Protocol

Syslog – System Logging Protocol

VM – Virtual Machine

CPU – Central Processing Unit

RAM – Random Access Memory

SSD – Solid State Drive

HDD – Hard Disk Drive

LAN – Local Area Network

IP – Internet Protocol

GUI – Graphical User Interface

SSH – Secure Shell

SIEM – Security Information and Event Management

URL – Uniform Resource Locator

LTS – Long Term Support

HTTP – Hypertext Transfer Protocol

RRD – Round Robin Database

API – Application Programming Interface

DNS – Domain Name System

OS – Operating System

CLI – Command Line Interface

MySQL – Structured Query Language Database Management System

LDAP – Lightweight Directory Access Protocol

CRS – Cloud Router Switch (від Mikrotik)

OID – Object Identifier

HTML – HyperText Markup Language

PHP – Hypertext Preprocessor

ВСТУП

Сучасні інформаційно-комунікаційні системи, особливо ті, що функціонують у критичних інфраструктурах, вимагають не лише стабільної роботи, але й постійного контролю за станом усіх мережевих компонентів. У зв’язку з цим питання побудови ефективної системи моніторингу стає ключовим як у контексті підтримання працездатності обладнання, так і в забезпеченні інформаційної безпеки. Наявність централізованого засобу спостереження за мережею дозволяє своєчасно виявляти аномалії, реагувати на інциденти та запобігати збоям, що можуть призвести до втрати даних або зниження рівня доступності сервісів.

У рамках цієї роботи розглянуто і реалізовано систему моніторингу мережевого середовища на основі програмного забезпечення з відкритим кодом – LibreNMS. Ця система зарекомендувала себе як потужний та масштабований інструмент, що підтримує велику кількість протоколів і типів обладнання, а також надає розвинені засоби інтеграції, візуалізації й оповіщення. Її впровадження дає змогу створити гнучке рішення, придатне до адаптації під специфіку конкретної інфраструктури.

Метою даного проєкту є розгортання та налаштування системи LibreNMS у віртуальному середовищі, проведення її тестування на прикладі реального обладнання, а також оцінка ефективності функціонування основних компонентів. У процесі роботи особливу увагу приділено інтеграції з протоколами SNMP, Syslog та ICMP, організації сповіщень про критичні події, візуалізації метрик і побудові централізованої системи логування. Отримані результати підтверджують доцільність застосування LibreNMS як універсального інструменту моніторингу, що відповідає сучасним вимогам до стабільності, безпеки та масштабованості мережевих рішень.

1 ВІДОМІ РІШЕНЯ МОНІТОРИНГУ МЕРЕЖЕВОЇ БЕЗПЕКИ

1.1 Загальні підходи до моніторингу мережевої інфраструктури

Мережевий моніторинг є невід'ємною складовою управління сучасною IT-інфраструктурою. Його головна функція полягає у спостереженні за станом мережевого обладнання, сервісів і каналів зв'язку з метою виявлення відхилень, збоїв і потенційних загроз. Збір даних здійснюється безперервно або періодично, що дозволяє оперативно реагувати на критичні події та підтримувати стабільність роботи мережі [1].

Основними завданнями моніторингу виступають контроль доступності ключових вузлів, аналіз продуктивності передавання даних, виявлення перевантажень та аномальної активності. Це забезпечує своєчасне виявлення точок відмови, мінімізацію простоїв і підтримку встановленого рівня сервісу [2].

Моніторинг також має важливу роль у контексті інформаційної безпеки. Системи фіксують підозрілі дії, несанкціоновані спроби доступу, зміну конфігурацій та відхилення від звичайного профілю трафіку. Таким чином він служить основою для раннього попередження про інциденти, допомагає в локалізації проблем і формує основу для аудиту подій у мережі.

До основних об'єктів мережевого моніторингу належать пристрої та компоненти, які формують функціональну основу інфраструктури. Серед них першочергову увагу приділяють кінцевим вузлам, маршрутизаторам, комутаторам і серверним системам. Ці елементи забезпечують обробку, зберігання і маршрутизацію трафіку, тому їхній стан безпосередньо впливає на доступність сервісів [3].

Моніторинг кінцевих точок дозволяє відстежувати активність користувачів і взаємодію з мережевими ресурсами. Контроль за комутаторами та маршрутизаторами забезпечує виявлення перевантажених портів, недоступних інтерфейсів або некоректної маршрутизації. Сервери перевіряються на рівні ресурсів, запущених служб і відповідей на мережеві запити.

Крім фізичних і логічних пристроїв, системи охоплюють канали передачі даних, ключові порти і мережеві сервіси. Йдеться про вимірювання пропускної

здатності, втрат пакетів, рівня затримок і стабільності з'єднань. Моніторинг портів дозволяє виявляти відкриті або небажані точки доступу, що є критично важливим для безпеки. Аналіз мережевих сервісів, таких як DNS, DHCP, HTTP або VPN, дає змогу оцінити їхню доступність та час реакції. Сукупний контроль усіх зазначених об'єктів створює повну картину стану інфраструктури та дозволяє оперативно виявляти проблеми на різних рівнях [4].

Методи збору інформації в системах моніторингу поділяються на кілька основних типів залежно від принципу взаємодії з мережевими об’єктами. Найпоширенішими є активний, пасивний та подієвий моніторинг. Кожен із цих підходів має свої переваги та обмеження, а ефективне рішення зазвичай поєднує кілька методів одночасно [5].

Активний моніторинг. Цей метод передбачає періодичне опитування пристроїв мережі з використанням протоколів SNMP, ICMP або API-запитів. Система ініціює запити до вузлів і аналізує отримані відповіді. Такий підхід дозволяє точно контролювати стан інтерфейсів, навантаження на процесор, використання оперативної пам’яті, обсяг трафіку на портах. Його головна перевага – стабільна регулярність збору даних і можливість детального налаштування частоти опитування. Однак надмірна кількість активних запитів може створювати додаткове навантаження на мережу [6].

Пасивний моніторинг. Здійснюється шляхом аналізу реального мережевого трафіку або журналів подій без прямого втручання в роботу пристроїв. Дані збираються через мережеві TAP-пристрої, дзеркальні порти або шляхом обробки логів з пристроїв. Метод дозволяє відстежувати фактичну поведінку систем, фіксувати підозрілі дії та виявляти аномалії у передачі даних. Він майже не створює навантаження на інфраструктуру, але потребує ефективних механізмів обробки великого обсягу даних [7].

Подієвий моніторинг. Заснований на отриманні повідомлень про події від мережевих пристроїв, таких як SNMP traps, syslog-записи або повідомлення про збої. У цьому випадку пристрій самостійно надсилає повідомлення до системи моніторингу у разі настання певної події. Цей метод забезпечує мінімальну затримку в отриманні критичної інформації, зменшує обсяг періодичних опитувань

і дозволяє швидко реагувати на інциденти. Водночас подієвий підхід потребує правильного налаштування відправників повідомлень і системи прийому [8].

Комплексне використання всіх трьох методів дозволяє досягти повного охоплення інфраструктури і забезпечити баланс між точністю, швидкістю реакції та навантаженням на мережу.

Протоколи моніторингу становлять основу взаємодії між системами збору даних і мережевими пристроями. Вони визначають правила доступу до інформації про стан інфраструктури, дозволяють здійснювати контроль за ключовими параметрами і забезпечують узгодженість між компонентами різних виробників [9].

Серед найпоширеніших протоколів у системах моніторингу використовується SNMP (Simple Network Management Protocol). Він дозволяє опитувати мережеві пристрої, отримувати значення різних метрик та реагувати на зміни у вигляді trap-повідомлень. Завдяки підтримці великою кількістю пристроїв SNMP є одним із базових механізмів активного моніторингу [10].

ICMP (Internet Control Message Protocol) застосовується для перевірки доступності вузлів, визначення затримок і втрат пакетів. Найбільш типовим прикладом є використання команд ping або traceroute. Протокол не вимагає автентифікації та дозволяє швидко визначати недоступні або нестабільні ресурси [11].

NetFlow і sFlow забезпечують детальний аналіз трафіку, що проходить через мережеві інтерфейси. Вони дозволяють виявляти джерела навантаження, нетипову активність і підозрілі шаблони передавання даних. Такі протоколи особливо цінні для виявлення аномалій, що можуть свідчити про спроби вторгнення або витоку даних [12].

WMI (Windows Management Instrumentation) використовується для збору даних з операційних систем Windows. Він дає змогу отримувати відомості про апаратне забезпечення, стан служб, процесів, оновлень та інші аспекти роботи серверів і клієнтських машин у середовищі Windows [13].

Syslog слугує універсальним каналом передавання журналів подій від пристроїв до централізованої системи зберігання і обробки логів. Він широко

підтримується в Linux/Unix-системах, маршрутизаторах, комутаторах та брандмауерах, забезпечуючи збирання повідомлень про стан, помилки, спроби доступу і зміну конфігурацій [14].

Таблиця 1.1 – Протоколи моніторингу

Протокол	Тип моніторингу	Призначення	Переваги	Недоліки
SNMP	Активний, подієвий	Опитування пристроїв, отримання метрик, traps	Широко підтримується, ефективний для базового моніторингу	Може мати затримки, потребує налаштування MIB
ICMP	Активний	Перевірка доступності, вимірювання затримок	Легкий у використанні, швидкий результат	Обмежена інформація, не завжди точний
NetFlow	Пасивний	Аналіз мережевого трафіку, виявлення аномалій	Детальний трафік-аналіз, ефективність для безпеки	Потребує додаткової обробки і зберігання даних
sFlow	Пасивний	Зразковий аналіз трафіку з низьким навантаженням	Менше навантаження на пристрої, масштабованість	Менш точний за NetFlow, залежить від реалізації
WMI	Активний	Отримання даних про системи Windows	Глибокий рівень даних для Windows-платформ	Не підтримується на інших ОС, складне налаштування
Syslog	Подієвий	Збирання логів подій з пристроїв і серверів	Стандартизований формат логів, централізація	Не завжди структуровані повідомлення, багато шуму
API	Активний, подієвий	Інтеграція з іншими системами, доступ до метрик	Гнучкість, автоматизація, розширюваність	Залежить від підтримки API, вимагає програмування

API-запити застосовуються для інтеграції систем моніторингу з іншими програмними компонентами. Через RESTful або інші типи API здійснюється доступ до метрик, керування пристроями та отримання сповіщень. Такий підхід є гнучким і розширює можливості автоматизації [15].

Використання стандартних, задокументованих протоколів забезпечує високу сумісність між різними компонентами інфраструктури. Це дозволяє об’єднувати обладнання різних виробників у єдину систему моніторингу без необхідності в глибокій адаптації. Уніфіковані протоколи також полегшують масштабування систем, зменшують складність інтеграції та спрощують подальшу підтримку.

Типи параметрів, що підлягають моніторингу в мережевій інфраструктурі, охоплюють кілька ключових категорій, кожна з яких має специфічне значення для стабільної та безпечної роботи системи. Моніторинг таких показників дає змогу своєчасно реагувати на відхилення, виявляти вузькі місця та забезпечувати відповідність політикам безпеки [16].

До параметрів продуктивності відносять показники, які відображають ефективність функціонування мережевих компонентів. Серед них – пропускна здатність каналів зв'язку, затримки при передачі даних і втрата пакетів. Аналіз цих величин дозволяє виявити перевантаження, неякісні канали або нестабільну маршрутизацію, що негативно впливають на якість сервісів [17].

Параметри доступності стосуються фактичної наявності та працездатності пристроїв і вузлів мережі. Вони включають стан мережевих інтерфейсів, результати ICMP-відгуків на запити ping та рівень аптайму. Постійне відстеження таких характеристик дозволяє оперативно виявляти відмови, визначати недоступні вузли та вчасно ініціювати відновлювальні дії [18].

Показники безпеки охоплюють події та активність, які можуть свідчити про спроби порушення цілісності чи конфіденційності системи. Йдеться про виявлення підозрілих з'єднань, спроб несанкціонованого доступу до вузлів, порушення політик автентифікації, а також зміни конфігурацій обладнання. Моніторинг цих параметрів є основою для виявлення атак, загроз і потенційно небезпечних дій всередині мережі [19].

Комплексне охоплення всіх трьох груп параметрів забезпечує повноцінну картину стану інфраструктури й дозволяє об’єктивно оцінювати як технічну справність, так і рівень захищеності мережі.

Моніторинг мережевої інфраструктури здійснюється на кількох рівнях мережевої моделі, зокрема на мережевому (L2/L3), транспортному (TCP/UDP) та прикладному (HTTP, DNS) рівнях. На рівні канального та мережевого рівнів аналізується структура трафіку відповідно до MAC- та IP-адрес, відстежуються зміни в таблицях ARP, виявляються конфлікти адрес, а також фіксується аномальна активність, пов'язана з підміною адрес, появою нових вузлів або змінами у маршрутизації. Збір інформації на цьому рівні дозволяє оперативно реагувати на такі загрози як ARP-spoofing, MAC-flooding, ICMP-флуд, а також забезпечує контроль за коректною роботою протоколів маршрутизації [20].

На транспортному рівні здійснюється аналіз TCP- і UDP-з'єднань, фіксується активність, пов'язана з відкритими портами, обробкою з'єднань та передачею даних між сервісами. Моніторинг цього рівня дозволяє виявляти аномалії, пов’язані зі спробами сканування портів, масовими підключеннями або раптовими змінами у поведінці мережевих сесій. Важливу роль відіграє також контроль за кількістю спроб встановлення з’єднань, перевищенням допустимої кількості запитів, а також нехарактерною активністю з боку окремих IP-адрес [21].

На прикладному рівні здійснюється аналіз специфічного трафіку, пов’язаного з роботою прикладних протоколів, таких як HTTP, HTTPS, DNS, FTP, SMTP та інші. У випадку з HTTP контролюється структура запитів, заголовки, параметри та частота звернень. Це дає змогу виявляти атаки типу SQL-ін'єкцій, спроби обхідних запитів, підозріле сканування веб-додатків та аномально високу активність. DNS-аналіз дозволяє виявити запити до підозрілих доменів, доменів, зареєстрованих нещодавно, або таких, що використовуються у фішингових кампаніях, а також фіксує ознаки DNS-тунелювання або DNS-ампліфікації [22].

Моніторинг на кожному з цих рівнів виконує взаємодоповнюючі функції і забезпечує комплексну оцінку стану мережевої безпеки. При правильній реалізації така багаторівнева система дозволяє не лише фіксувати поточні загрози, а й виявляти тенденції, прогнозувати атаки і формувати ефективні сценарії реагування.

Централізовані системи моніторингу відіграють ключову роль у забезпеченні цілісності, узгодженості та оперативності збору даних про стан мережевої інфраструктури. На відміну від розподілених підходів, де кожен вузол може вести облік лише локальних подій, централізована система забезпечує єдину точку агрегації, аналізу та кореляції подій з усіх компонентів мережі. Це дозволяє формувати повну картину мережевої активності в реальному часі, відстежувати тренди, виявляти закономірності та порушення у поведінці трафіку [23].

Збір метрик з різних рівнів – від фізичних інтерфейсів до прикладних служб – потребує уніфікованого механізму доставки даних. Централізовані системи зазвичай базуються на використанні агентів, протоколів типу SNMP, NetFlow, sFlow або сучасних telemetry-рішень, що дозволяє отримувати як періодичні статистичні дані, так і потокову інформацію з високою частотою оновлення. Отримані метрики включають навантаження на канали, затримки, кількість з'єднань, типи протоколів, частоту помилок, а також специфічні показники безпеки, зокрема спроби доступу, зміни конфігурацій, сигнатури атак [24].

Обробка цих даних в централізованій системі базується на використанні механізмів нормалізації, фільтрації, кореляції та збагачення інформації з додаткових джерел, таких як бази загроз, геолокаційні сервіси або історичні архіви логів. Це дає змогу не лише фіксувати окремі інциденти, а й виявляти складні багаторівневі атаки, приховану активність або аномальні взаємодії між сегментами мережі. Окрім цього, централізоване зберігання метрик забезпечує історичний аналіз, порівняння з базовими значеннями та підготовку звітності відповідно до політик безпеки.

Важливою функцією таких систем є автоматизація реагування: на основі заздалегідь визначених політик або моделей машинного навчання система може ініціювати сповіщення, ізоляцію вузлів, зміни в конфігураціях або запуск скриптів протидії. Таким чином, централізований підхід дозволяє не лише збирати дані, а й активно управляти станом безпеки в масштабах усієї інфраструктури.

Ефективна інтеграція системи моніторингу з політикою інформаційної безпеки передбачає не лише пасивне спостереження за станом мережі, а й активне виявлення інцидентів шляхом аналізу аномальної поведінки. Такий підхід

ґрунтується на попередньо визначених моделях "нормального" функціонування інфраструктури, які формуються на основі статистичних метрик, історичних даних та поточних шаблонів доступу до мережевих ресурсів. Відхилення від цих моделей фіксуються системою моніторингу як потенційні індикатори компрометації.

Аномалії можуть проявлятись у вигляді різкого зростання обсягу вхідного або вихідного трафіку, зміни географії підключень, нетипового використання портів, сплесків DNS-запитів до невідомих доменів, а також нетипових запитів до внутрішніх сервісів у нестандартний час. Часто такі відхилення вказують на активну фазу атаки або вторгнення – зокрема сканування мережі, спроби вертикального або горизонтального переміщення, ексфільтрацію даних або зловмисне використання внутрішніх ресурсів [25].

Системи, що підтримують поведінковий аналіз, можуть автоматично порівнювати поточну активність із типовими шаблонами, застосовуючи порогові значення, кореляційні правила або алгоритми машинного навчання. Таким чином забезпечується динамічне виявлення подій, які не були визначені як загрози на рівні сигнатур. Це особливо актуально у випадках нульового дня або складних атак, які обходять традиційні механізми фільтрації [26].

Інтеграція з безпековою політикою полягає в узгодженні виявлених аномалій із забороненими сценаріями доступу, правилами сегментації, політиками автентифікації та журналювання. У результаті інциденти не лише фіксуються, а й відразу класифікуються за рівнем ризику відповідно до внутрішніх норм. Це дозволяє швидко приймати рішення про ізоляцію підозрілих вузлів, обмеження доступу, початок розслідування або повідомлення відповідальних осіб. Таким чином, виявлення інцидентів через поведінкові аномалії стає важливим елементом системи активного захисту, що діє у рамках загальної безпекової стратегії організації.

Сучасні системи моніторингу безпеки орієнтовані не лише на виявлення загроз, а й на миттєву реакцію на них відповідно до визначеної безпекової політики. Реагування у реальному часі передбачає автоматичне виконання дій при виявленні подій, що відповідають ознакам інциденту. Такі дії можуть бути як превентивними,

так і коригувальними: блокування з’єднань, обмеження доступу, переведення інтерфейсів у захищений режим, ізоляція хостів або активація скриптів реагування.

Для реалізації цієї функціональності система має бути здатна отримувати дані з різних джерел з мінімальною затримкою, обробляти їх в контексті заздалегідь визначених правил або моделей поведінки і одразу ініціювати відповідні дії через інтегровані канали управління. Прикладом є зв’язок між системою виявлення вторгнень і міжмережевим екраном, де при фіксації атаки на рівні прикладного протоколу одразу змінюється політика фільтрації трафіку.

Ключовим є не лише швидкість реакції, а й її узгодженість із загальною політикою безпеки. Автоматизовані дії повинні бути контрольованими, прогнозованими та логованими, щоб уникнути хибнопозитивних блокувань або втрати доступності критичних сервісів. У складніших сценаріях ініціюється напівавтоматичний режим – система повідомляє відповідального фахівця, пропонує варіанти дій, і лише після підтвердження виконує реакцію.

Крім захисних заходів, реакція у реальному часі включає оновлення індикаторів компрометації, коригування порогів спрацьовування, а також передавання даних до SIEM-систем або центрів реагування на інциденти. Усе це забезпечує безперервний захист інфраструктури, скорочує час між виявленням та нейтралізацією загрози і знижує потенційні наслідки атак або внутрішніх порушень.

1.2 Класифікація та функціональні можливості систем моніторингу

Системи моніторингу мережевої інфраструктури класифікуються за різними критеріями залежно від архітектури, джерел даних, рівнів контролю та інтеграційних можливостей. Одним з основних критеріїв є тип джерела, з якого отримується інформація. Агентські системи використовують встановлені на хостах програми для збору локальних метрик, тоді як безагентські рішення працюють за рахунок опитування протоколами SNMP, NetFlow або зчитування журналів з

мережевих пристроїв. Гібридні підходи поєднують обидва методи, забезпечуючи гнучкість у виборі джерел і рівнів доступу [27].

За принципом дії системи моніторингу поділяються на активні та пасивні. Активні здійснюють контроль стану об'єктів через періодичне надсилання запитів, наприклад, ICMP, SNMP або HTTP-запити, що дозволяє оцінити доступність і швидкодію. Пасивні системи, навпаки, аналізують трафік без втручання в нього, використовуючи дзеркалювання портів або додаткові пристрої для зчитування потокових даних. Комбіновані системи використовують обидва підходи, забезпечуючи як швидке виявлення відмов, так і глибокий аналіз трафіку [28].

Типізація систем також враховує масштаб і призначення. Локальні рішення застосовуються в межах окремих сегментів або об'єктів. Корпоративні системи охоплюють всю організаційну інфраструктуру, підтримуючи декілька підмереж, регіонів і рівнів доступу. Хмарні сервіси надають гнучкість і масштабованість, дозволяючи інтегрувати моніторинг у хмарні середовища. Телеметричні рішення, що базуються на потоковій передачі даних, орієнтовані на високу щільність збору метрик у реальному часі [29].

Архітектура системи також визначає її функціональність. Централізовані системи збирають і обробляють інформацію у єдиному центрі, що спрощує адміністрування та кореляцію подій. Децентралізовані підходи використовуються у великих розподілених середовищах, де окремі вузли самостійно здійснюють первинну обробку даних. Розподілені системи зберігають баланс між автономністю окремих компонентів та централізованою координацією аналітики.

Важливим аспектом є інтеграція систем моніторингу з існуючою інфраструктурою. Ефективні рішення повинні підтримувати широкий перелік обладнання, протоколів збору даних, API для обміну інформацією, а також бути сумісними з мережевими стандартами. Це забезпечує повноцінне охоплення контрольованих об'єктів і мінімізує потребу у ручній конфігурації.

Системи моніторингу повинні підтримувати аналіз мережевого трафіку на всіх рівнях – від канального до прикладного. Це дає змогу контролювати як фізичні параметри каналів, так і логіку взаємодії між клієнтом і сервером, включаючи

специфіку протоколів HTTP, DNS, SMTP тощо. Такий підхід забезпечує глибоку інспекцію трафіку та детальне розуміння поведінки мережі [30].

Однією з ключових функцій є виявлення аномалій і інцидентів безпеки. У цьому контексті застосовуються сигнатурні методи, що порівнюють поточні події з базою відомих загроз, а також поведінковий аналіз, що виявляє відхилення від типових шаблонів. Аналітичні методи з використанням машинного навчання дозволяють розпізнавати складні загрози на основі контексту та кореляції подій.

Функціональність реагування включає можливість ініціювати дії у відповідь на загрозу. Це може бути блокування трафіку, зміна конфігурації, сповіщення відповідальних осіб або активація політик безпеки. Автоматизація цих процесів підвищує швидкість реагування і зменшує навантаження на операторів.

Система повинна бути здатна до збору великих обсягів метрик у режимі реального часу, агрегації показників та зберігання історичних даних для подальшого аналізу. Це передбачає використання оптимізованих сховищ, механізмів стиснення та індексації, що дозволяє забезпечити як швидкодію, так і масштабованість.

Для ефективного прийняття рішень потрібна наочна візуалізація зібраних даних. Важливими елементами є дашборди, інтерактивні графіки, топологічні карти, які дозволяють оператору миттєво оцінити стан мережі та виявити критичні відхилення [31].

Системи повинні мати можливість масштабування відповідно до зростання навантаження – як за кількістю контрольованих вузлів, так і за обсягом трафіку. Підтримка кластеризації, розподілених баз даних та балансування навантаження є критичними для великих розгортань.

Інтеграція з іншими інструментами кібербезпеки, зокрема з SIEM-системами, системами виявлення та запобігання вторгненням (IDS/IPS), а також системами контролю доступу, дозволяє будувати єдине середовище контролю і забезпечувати комплексну безпеку [32].

Дотримання вимог політик безпеки та нормативних стандартів (наприклад, ISO/IEC 27001, GDPR) є обов’язковим для впровадження у критичних

середовищах. Система повинна підтримувати відповідні засоби контролю доступу, сегментації, логування і звітності [33].

Наявність механізмів аудиту та журналювання дозволяє фіксувати всі дії користувачів і зміни в системі, що є необхідною умовою для розслідування інцидентів та підготовки звітів для внутрішнього або зовнішнього аудиту. Автоматичне формування звітів спрощує контроль дотримання політик та планування заходів з підвищення безпеки.

Застосування алгоритмів машинного навчання дозволяє системі адаптуватися до змін у поведінці трафіку, самостійно оновлювати моделі нормального функціонування та підвищувати точність виявлення нових типів загроз. Це розширює функціональні можливості традиційного моніторингу і підвищує його ефективність у складних середовищах.

1.3 Огляд відомих систем моніторингу

Zabbix – це потужна система моніторингу корпоративного рівня, яка забезпечує збір, зберігання, аналіз і візуалізацію даних у реальному часі. Вона підтримує моніторинг широкого спектру ресурсів, включаючи сервери, мережеве обладнання, додатки, служби, БД, віртуалізацію та хмарні сервіси. Основою архітектури є агентське та безагентське спостереження, що дозволяє здійснювати гнучке налаштування точок збору метрик через SNMP, IPMI, JMX, HTTP, SSH, WMI, а також сторонні інтеграції. Zabbix дозволяє створювати детальні шаблони моніторингу з наборами елементів даних, тригерів, візуалізацій і сценаріїв реакцій, що прискорює розгортання та стандартизує контроль за станом інфраструктури [34].

Сильними сторонами системи є висока масштабованість, можливість обробки сотень тисяч метрик у розподіленому середовищі, потужна система сповіщень з багатьма каналами доставки (email, webhook, Telegram, Slack тощо) та умовною логікою для гнучкого керування алертами. Веб-інтерфейс Zabbix надає глибоку візуалізацію стану системи, включаючи графіки, heatmaps, top-N види, а

також дозволяє будувати дашборди, звіти і переглядати історію подій. Підтримується автоматичне виявлення хостів і сервісів, що полегшує управління великими динамічними середовищами.

До слабких сторін можна віднести складність початкового налаштування, особливо в частині розгортання великомасштабної системи з високим навантаженням, потребу в глибокому розумінні моделі даних та правил тригерів, а також громіздкий інтерфейс для роботи з великою кількістю хостів або шаблонів. Деякі функції, такі як керування конфігурацією через API або масштабування за допомогою проксі, потребують додаткового адміністрування і документації. Крім того, Zabbix не має вбудованої підтримки сучасних DevOps-підходів, таких як інтеграція з Kubernetes, Prometheus чи Grafana, хоча частково ці можливості доступні через сторонні інтеграції.

Вікно відображення роботи Zabbix представлено на рис. 1.1. У цілому Zabbix підходить для організацій, яким потрібне централізоване, надійне та гнучке рішення для моніторингу з великими можливостями кастомізації та розширення, проте воно вимагає часу та ресурсів на розгортання, налаштування та підтримку.

Рисунок 1.1 – Вікно відображення роботи Zabbix

LibreNMS – це автоматизована система моніторингу мережевих пристроїв з відкритим кодом, яка орієнтована переважно на SNMP-базоване спостереження за станом комутаторів, маршрутизаторів, бездротових точок доступу, серверів, а

також деяких типів систем з API-доступом. Система підтримує автоматичне виявлення пристроїв, оновлення інтерфейсів, VLAN, портів і зв'язків, що спрощує її інтеграцію у динамічні середовища. LibreNMS має розгалужену підтримку MIB-базованих пристроїв, здатна збирати тисячі показників продуктивності з різних виробників без потреби у складному конфігуруванні шаблонів [35].

Однією з ключових можливостей є інтуїтивно зрозумілий веб-інтерфейс з відображенням метрик у вигляді графіків, таблиць, статусів портів, використання пропускної здатності, температур, стану системи тощо. Система дозволяє будувати сповіщення з широким набором умов, логіки затримки і підтвердження, а також підтримує безліч каналів повідомлень, зокрема email, Slack, Telegram, Discord, Microsoft Teams, webhook. Важливою є підтримка розмежування доступу, мульти-юзерного керування, REST API для автоматизації інтеграцій і підтримка зовнішніх скриптів через модулі транспортів. LibreNMS сумісна з RANCID, Oxidized, NFSen, Collectd, а також може експортувати дані в InfluxDB для подальшої обробки в Grafana.

Сильними сторонами LibreNMS є простота розгортання, активна підтримка спільноти, регулярні оновлення, широке покриття пристроїв завдяки SNMP та Auto-Discovery, повна підтримка візуалізації історії метрик з можливістю детального аналізу. Інтеграція з Oxidized дозволяє зберігати бекапи конфігурацій мережевого обладнання, що додає базову функцію контролю версій. LibreNMS добре працює у середовищах з великою кількістю однотипних пристроїв, таких як ISP, дата-центри або університетські мережі.

Слабкими сторонами є обмежена можливість глибокого моніторингу серверних сервісів або додатків, оскільки основна увага приділена мережевим пристроям. Візуалізація, хоча й зручна, не має гнучкості у побудові дашбордів порівняно з системами типу Zabbix або Grafana. Масштабування у великих середовищах потребує оптимізації бази даних і часом нестабільне без належної підтримки. Високе навантаження на базу даних RRD або MySQL при великій кількості пристроїв може впливати на швидкість відповіді інтерфейсу. Також LibreNMS не має повноцінної моделі шаблонів з групуванням метрик або кастомним спадкуванням, що обмежує повторне використання конфігурацій між

пристроями. Окремим викликом є відсутність вбудованих засобів для складного корелювання подій чи логічної обробки залежностей між пристроями.

Вікно відображення роботи LibreNMS представлено на рис. 1.2. У підсумку, LibreNMS – ефективне рішення для автоматизованого SNMP-моніторингу з мінімальним порогом входу, високим рівнем автоматизації та хорошим покриттям мережевої інфраструктури, але з обмеженою функціональністю у сфері глибокого моніторингу серверних застосунків, обробки залежностей та складної кастомізації.

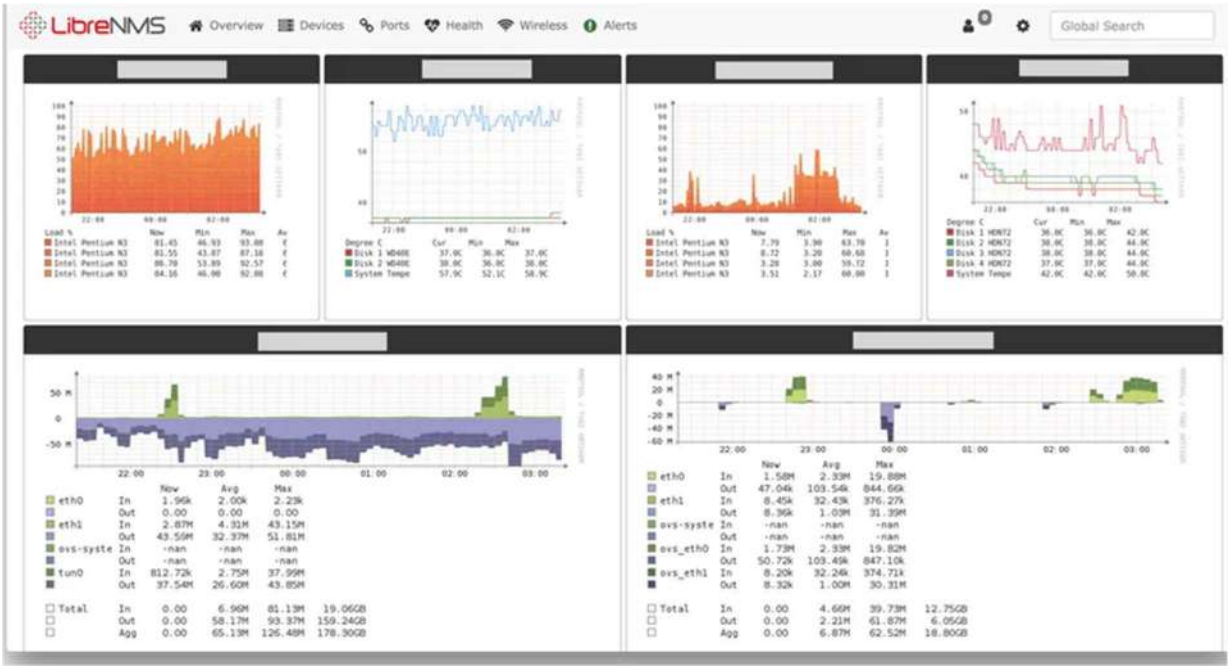


Рисунок 1.2 – Вікно відображення роботи LibreNMS

Nagios – це одна з найстаріших і найвідоміших систем моніторингу, яка зосереджена на контролі доступності та працездатності серверів, сервісів, мережевого обладнання та інфраструктурних компонентів. Архітектура Nagios побудована навколо центрального ядра з модульною системою плагінів, які виконують перевірки за допомогою зовнішніх скриптів або команд. Вона використовує активний або пасивний моніторинг із періодичним опитуванням хостів, збором відповідей на перевірки та генерацією подій, які потім обробляються системою сповіщень. Nagios підтримує широке розширення функцій через власні або сторонні плагіни, що дозволяє адаптувати систему до практично будь-яких типів перевірок, від пінгу до складної логіки аналізу даних сервісів чи журналів [36].

Сильними сторонами Nagios є його гнучкість, висока надійність у базових сценаріях моніторингу, проста логіка перевірок, велика кількість існуючих плагінів, а також підтримка стандартів оповіщення, як-от email або SMS через шлюзи. Крім того, завдяки своїй популярності, Nagios має потужну екосистему – численні розширення, інтерфейси візуалізації, API, а також похідні проєкти, зокрема Nagios XI, Icinga, Naemon, які додають більше функціональності або сучасніший інтерфейс. Система добре підходить для малих і середніх середовищ, де потрібен жорсткий контроль доступності без необхідності збору великої кількості метрик у реальному часі.

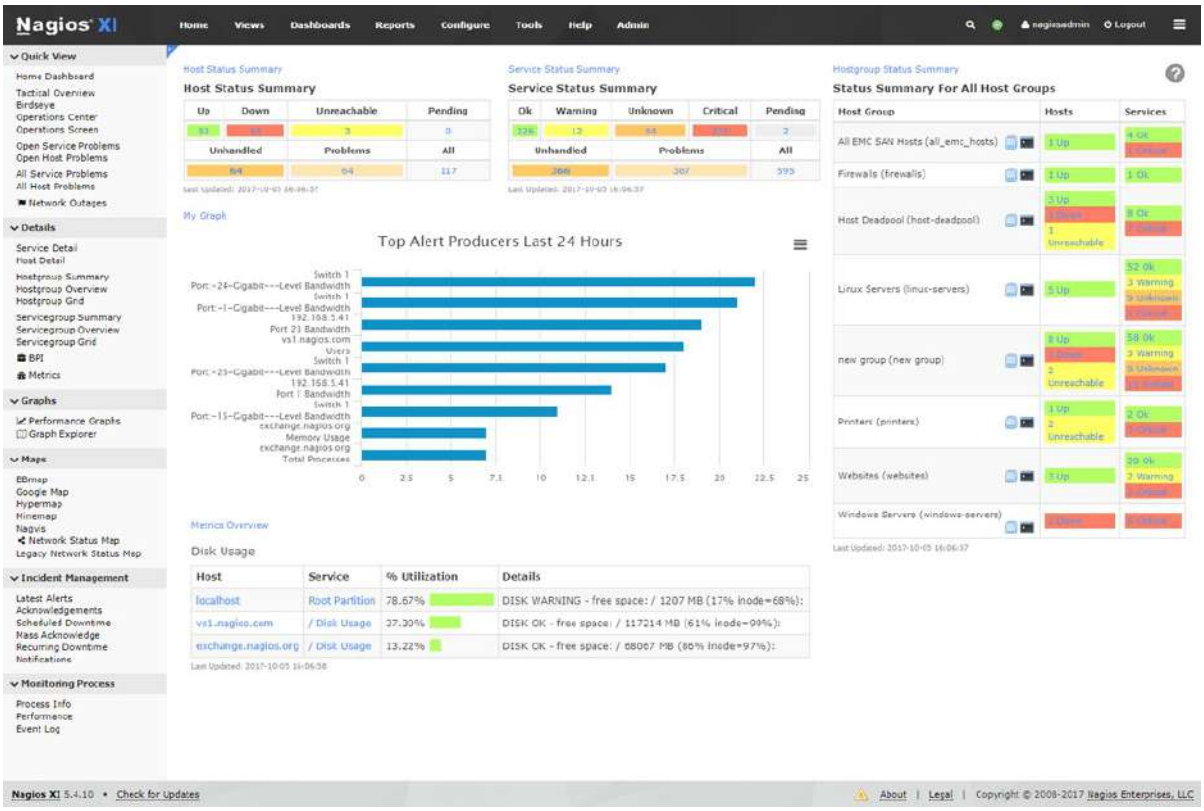


Рисунок 1.3 – Вікно відображення роботи Nagios

До слабких сторін належить складність конфігурації, оскільки базовий Nagios вимагає ручного створення текстових конфігурацій без централізованої системи шаблонів або автодетекції. Відсутність вбудованої візуалізації метрик, слабка підтримка історичних даних і обмежений інтерфейс керування роблять систему менш зручною порівняно з сучаснішими рішеннями. Масштабування вимагає значного налаштування із використанням розподілених компонентів або

сторонніх рішень. Крім того, Nagios орієнтований переважно на статусний моніторинг – він повідомляє, коли щось перестає працювати, але не забезпечує широкого інструментарію для аналізу продуктивності, трендів або кореляції подій. Його модель подій не підтримує складної логіки залежностей або умов спрацювання без додаткового розширення. У сучасному DevOps-середовищі Nagios часто потребує інтеграції з іншими системами для отримання повноцінної картини – наприклад, з InfluxDB і Grafana для зберігання й візуалізації метрик або з Elastic Stack для обробки логів [38].

Вікно відображення роботи Nagios представлено на рис. 1.3. Загалом, Nagios залишається надійною платформою для базового моніторингу доступності з гнучкою архітектурою, проте її функціональність і зручність поступаються сучасним комплексним системам, які поєднують моніторинг, аналітику, алертинг і автоматизацію у єдиній платформі.

PRTG Network Monitor – це комплексна комерційна система моніторингу, розроблена компанією Paessler, яка орієнтована на просте впровадження, візуальний контроль і централізований моніторинг інфраструктури. Основна концепція системи побудована навколо сенсорів – кожен сенсор відповідає за окремий параметр або сервіс, наприклад, доступність хосту, завантаження процесора, трафік інтерфейсу, відповідь HTTP, стан служби Windows або дані з SNMP. PRTG підтримує широкий набір протоколів і технологій, зокрема SNMP, WMI, NetFlow, sFlow, IP SLA, Packet Sniffing, REST API, PowerShell, а також дозволяє створювати власні скриптові сенсори для інтеграції з нестандартними рішеннями [39].

Система має зручний графічний інтерфейс з динамічними дашбордами, групуванням пристроїв за ієрархією, кольоровим кодуванням статусів, а також багаторівневою візуалізацією від пристрою до окремого сенсора. Налаштування сповіщень у PRTG побудоване на базі гнучких тригерів з умовами, затримками та повторенням, і може здійснюватися через email, SMS, push-повідомлення або зовнішні скрипти. Вбудована система звітів дозволяє формувати як регулярну звітність, так і експрес-аналіз поточного стану. Підтримується інтеграція з Active Directory, а також існують мобільні додатки для Android та iOS для дистанційного

спостереження. Окремо варто відзначити підтримку кластеризації для високої доступності та масштабованості системи [40].

Серед сильних сторін PRTG варто виділити швидке розгортання, автоматичне виявлення пристроїв і сервісів, сучасний інтерфейс користувача, хорошу візуалізацію і низький поріг входу для адміністратора без глибоких знань у сфері моніторингу. Проста модель налаштування через GUI дозволяє швидко створювати нові сенсори, групи, шаблони. Висока інтеграція з Microsoft-орієнтованими інфраструктурами робить систему особливо привабливою для середовищ, де домінують Windows-сервери. Офіційна підтримка, документація та велика база знань значно спрощують процес експлуатації.

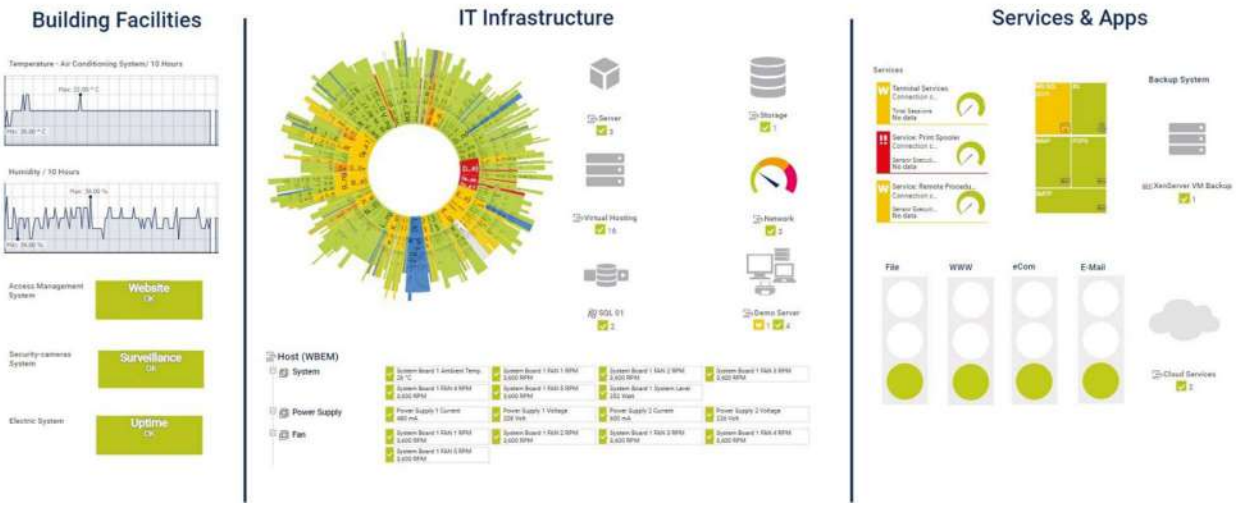


Рисунок 1.4 – Вікно відображення роботи PRTG Network Monitor

До слабких сторін можна віднести те, що система є пропрієтарною і повноцінне використання передбачає платну ліцензію, що обмежує її застосування в середовищах з великою кількістю сенсорів. Ліцензування базується на кількості сенсорів, що змушує ретельно планувати кожен елемент моніторингу. Обмеження у кастомізації логіки оповіщень або побудови складних залежностей безпосередньо в GUI також можуть стати стримуючим фактором у складних середовищах. Крім того, система переважно працює у Windows-середовищі, що робить її менш зручною для Linux-центричних інфраструктур. Масштабування системи на тисячі сенсорів потребує ретельного планування ресурсів сервера, а продуктивність

сильно залежить від потужності хост-машини. Відсутність відкритого коду і обмеження в гнучкості плагінів робить PRTG менш привабливим для DevOps- або хмарно-орієнтованих сценаріїв, де необхідна глибока інтеграція з CI/CD, хмарними API та контейнерною інфраструктурою [41].

Вікно відображення роботи PRTG Network Monitor представлено на рис. 1.4. Загалом PRTG – це потужне рішення для мережевого та інфраструктурного моніторингу у середовищах малого й середнього масштабу з високою візуалізацією та простотою налаштування, однак з деякими обмеженнями масштабованості, гнучкості і відкритості порівняно з більш кастомізованими системами.

1.4 Порівняльний аналіз систем за критеріями безпеки та масштабування

З метою забезпечення порівняння розглянутих систем оберемо критерії їх порівняння. Ці критерії охоплюють основні аспекти оцінки систем моніторингу. Архітектура та принципи роботи визначають, як система побудована та які моделі використовує (агентська, безагентська, гібридна). Підтримувані технології та протоколи моніторингу показують сумісність із різними джерелами даних – SNMP, WMI, NetFlow, API тощо [42]. Інтерфейс користувача та візуалізація відображають зручність перегляду й аналізу даних через дашборди, графіки й статусні панелі. Гнучкість у налаштуванні й кастомізації стосується можливості змінювати шаблони, тригери, логіку сповіщень. Автоматичне виявлення пристроїв і сервісів дозволяє мінімізувати ручну конфігурацію при розгортанні [43]. Механізми сповіщення та управління подіями оцінюють, наскільки гнучко система реагує на події та критичні ситуації. Масштабованість і продуктивність описують, як система працює у великих мережах і під навантаженням. Безпека охоплює захист доступу, аутентифікацію, контроль прав, шифрування та аудит дій. Підтримка інтеграцій і API визначає відкритість до взаємодії з іншими інструментами та системами. Можливості автоматизації й DevOps-сумісність оцінюють здатність системи інтегруватися з CI/CD, хмарними або контейнеризованими рішеннями [44]. Модель ліцензування та вартість володіння важливі з точки зору економіки впровадження.

Підтримка користувачів, документація та оновлення відображають якість офіційної підтримки, активність розробників і доступність навчальних матеріалів.

Таблиця 1.2 – Критерії оцінки систем

Критерії\Системи	LibreNMS	Zabbix	Nagios	PRTG
Архітектура та принципи роботи	5	4	3	4
Підтримувані технології та протоколи	5	4	3	5
Інтерфейс користувача та візуалізація	4	3	2	5
Гнучкість у налаштуванні й кастомізації	5	3	4	3
Автоматичне виявлення пристроїв і сервісів	5	5	2	5
Механізми сповіщення та управління подіями	5	4	3	4
Масштабованість і продуктивність	5	3	2	4
Безпека доступу та захист даних	4	3	3	4
Підтримка інтеграцій і розширюваність API	5	4	3	3
Автоматизація та DevOps-сумісність	3	2	2	2
Модель ліцензування та вартість володіння	5	5	5	2
Підтримка, документація та оновлення	4	4	4	5

1.5 Постановка задачі

Проведений аналіз існуючих рішень для моніторингу мережевої інфраструктури показав, що оптимальним рішенням є використання системи LibreNMS, яка забезпечує автоматичний моніторинг стану мережевого обладнання на основі SNMP, підтримує автоматичне виявлення пристроїв, має зручний веб-інтерфейс та достатньо високий рівень безпеки й гнучкості. Для ефективного контролю та управління мережею необхідно налаштувати систему моніторингу LibreNMS, яка має відповідати таким вимогам:

– забезпечити автоматичне виявлення та моніторинг усіх вузлів мережі, включаючи маршрутизатори, комутатори, сервери, точки доступу Wi-Fi та камери відеоспостереження;

– реалізувати систему сповіщень через електронну пошту;

– налаштувати розмежування прав доступу до інтерфейсу моніторингу відповідно до ролей користувачів;

– забезпечити постійний аналіз продуктивності та завантаження мережевих вузлів з можливістю зберігання історичних даних для аудиту подій та аналізу аномалій;

– створити необхідні графіки та звіти для оперативного реагування на інциденти й оцінки стану мережі.

Результатом роботи повинно бути повністю налаштоване і працездатне рішення на базі LibreNMS, яке забезпечить якісний моніторинг та відповідність встановленим політикам інформаційної безпеки мережі.

2 LIBRENMS ЯК ОСНОВА СИСТЕМИ МОНІТОРИНГУ БЕЗПЕКИ

2.1 Архітектура LibreNMS та її функціональні можливості

Загальна архітектура LibreNMS побудована за класичною клієнт–серверною схемою і передбачає чітке розмежування функціональних компонентів: веб-інтерфейсу, системи збору даних (полерів), сховища даних та механізмів масштабування. Центральним елементом є сервер з веб-інтерфейсом, реалізованим на базі PHP-фреймворку Laravel з використанням Apache або Nginx як веб-сервера, що забезпечує графічний доступ до налаштувань, графіків і панелей моніторингу [46].

Збір інформації з мережевих пристроїв виконується за допомогою окремих процесів (pollers), які здійснюють регулярні опитування пристроїв, використовуючи переважно SNMP-протокол. Ці процеси можуть працювати як на основному сервері, так і бути розподіленими по кількох вузлах, формуючи розподілену архітектуру, яка дозволяє ефективно масштабувати систему. Poller-и виконують не лише простий збір даних, а й активне виявлення (auto-discovery) нових пристроїв та інтерфейсів у мережі, автоматично оновлюючи конфігурацію системи.

Отримані дані зберігаються у двох типах сховищ: основна інформація, конфігурації пристроїв, налаштування, події та користувачі розміщуються у реляційній базі MySQL (MariaDB), тоді як метрики часу та дані продуктивності зберігаються у форматі Round Robin Database (RRD), який забезпечує компактне зберігання та швидкий доступ до історичних даних. Доступ до RRD може бути додатково оптимізований за допомогою RRDcached [47].

Для забезпечення узгодженості роботи кількох паралельних poller-процесів LibreNMS використовує механізми синхронізації через Redis або Memcached, що дозволяє уникнути конфліктів і забезпечити ефективне керування ресурсами у випадку масштабування. Архітектура LibreNMS також підтримує горизонтальне масштабування завдяки розподіленим poller-ам, які спільно використовують загальні ресурси (бази даних, сховища RRD), забезпечуючи баланс навантаження та підвищуючи загальну відмовостійкість системи [48].

Таким чином, архітектура LibreNMS дозволяє гнучко масштабувати мережевий моніторинг, ефективно обслуговувати велику кількість пристроїв, та забезпечувати високий рівень доступності й надійності даних завдяки модульності, чіткій структурі компонентів і можливості розподілу навантаження рис. 2.1.

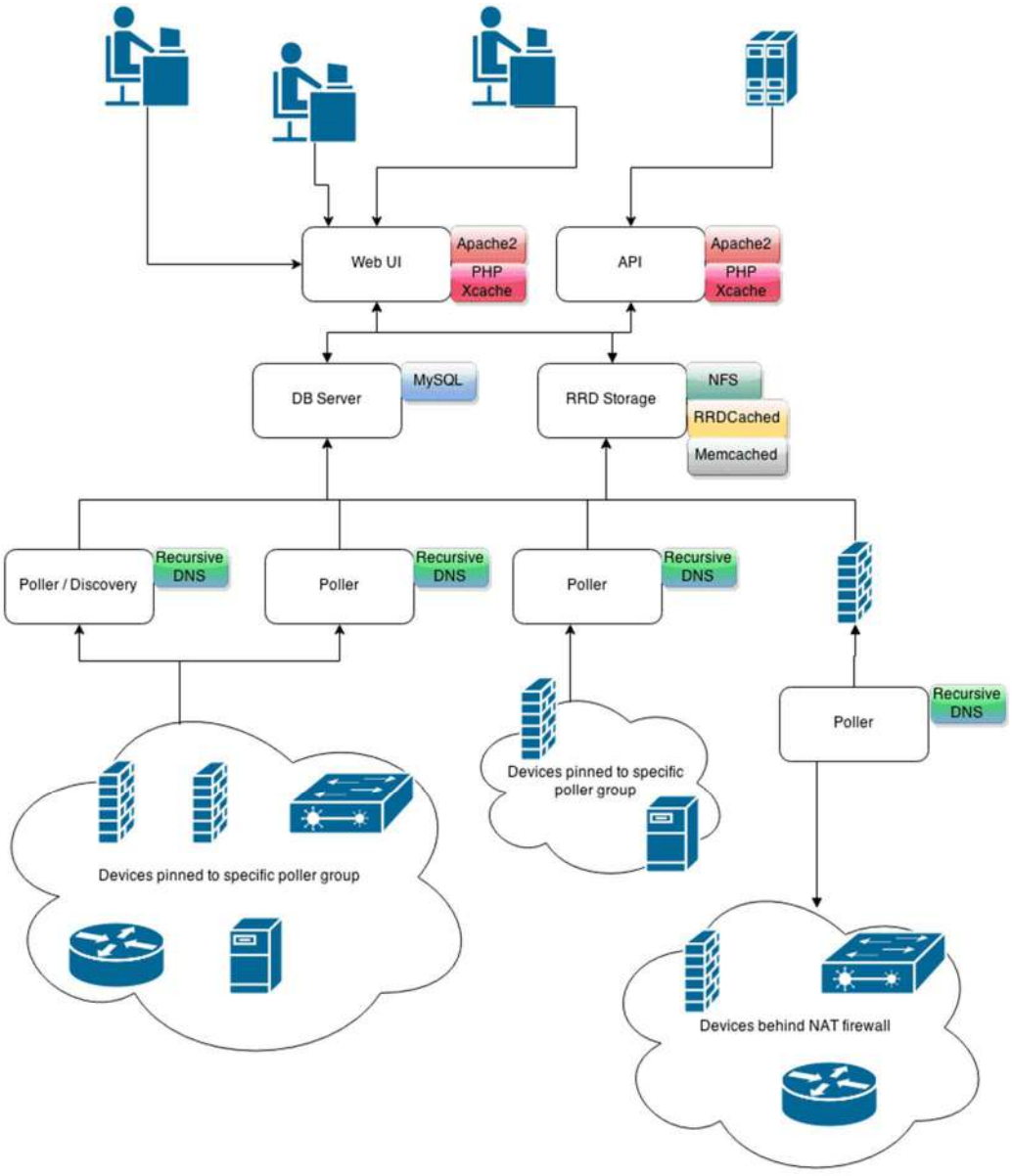


Рисунок 2.1 – Архітектура LibreNMS

LibreNMS підтримує широкий набір сучасних технологій та протоколів, що дозволяє йому ефективно взаємодіяти з різними мережевими пристроями та забезпечувати всебічний моніторинг мережевої інфраструктури.

Основні підтримувані технології включають:

– SNMP (Simple Network Management Protocol) – базова технологія для збору інформації з пристроїв.

– Auto-Discovery – автоматичне виявлення нових пристроїв та оновлення конфігурації.

– Syslog – збір журналів подій для аналізу та реагування на інциденти.

– NetFlow/sFlow – аналіз трафіку та виявлення аномалій на рівні мережевих потоків.

– LLDP/CDP (xDP) – автоматичне визначення топології мережі.

– REST API – забезпечення інтеграції та автоматизації роботи системи.

– Oxidized – зберігання резервних копій конфігурацій пристроїв з керуванням версіями.

– InfluxDB – зберігання часових рядів даних для подальшого аналізу і візуалізації (зокрема через Grafana).

Підтримувані протоколи: SNMP, ICMP, LLDP, CDP, NetFlow, sFlow, Syslog, HTTP, HTTPS, SMTP, SSH.

Завдяки такому широкому спектру підтримуваних технологій і протоколів LibreNMS здатен комплексно контролювати стан і продуктивність мережевого обладнання різних виробників та типів, здійснювати глибокий аналіз поведінки мережі, своєчасно виявляти проблеми та аномалії і оперативно реагувати на інциденти [49].

LibreNMS автоматично виявляє пристрої, використовуючи низку методів: ARP, SNMP Scan, LLDP/CDP/OSPF/BGP та сканування IP-діапазонів через snmp-scan.py. Після налаштування SNMP-спільнот і дозволених мереж кожні 6 годин запускається процес Auto-Discovery, який знаходить нові пристрої чи інтерфейси і додає їх до бази. Poller-и потім регулярно опитують ці пристрої за SNMP, збираючи метрики CPU, пам’яті, інтерфейсів, температури та інших параметрів. Зібрані дані зберігаються в базі MySQL (конфігурації, події) та RRD (часові метрики), що забезпечує історичний аналіз. Крім цього, LibreNMS підтримує ARP-і XDP-виявлення для побудови топології мережі й автоматичного створення мережевих

карт. Опціонально можна налаштувати snmp-scan.py в кроні з параметром -P для сканування IP-діапазонів і виявлення пристроїв, що відповідають. Цей підхід забезпечує майже повне охоплення мережі та автоматизацію налаштування моніторингу без потреби ручного додавання кожного пристрою рис 2.2.

Рисунок 2.2 – Автоматичний збір метрик

LibreNMS має розвинену систему сповіщень, яка дозволяє оперативно інформувати адміністраторів та операторів про критичні події в мережі. Вона підтримує широке різноманіття каналів зв'язку, включаючи електронну пошту, Telegram, Slack, Discord, Microsoft Teams, SMS, і Webhooks, що дає змогу легко інтегруватися з існуючими процесами реагування. Конфігурація повідомлень є гнучкою і дозволяє визначати різноманітні умови спрацювання тригерів, такі як недоступність пристроїв, перевантаження ресурсів або аномальна поведінка [50].

Безпека доступу забезпечується через розмежування прав користувачів і інтеграцію з зовнішніми системами авторизації. LibreNMS підтримує авторизацію через LDAP та Active Directory, що дозволяє централізовано управляти обліковими записами та доступом до системи відповідно до визначених ролей. Це суттєво знижує ризики несанкціонованого доступу та забезпечує чіткий контроль за доступом до конфігурацій і моніторингових даних.

Управління подіями реалізується шляхом централізованого збирання і класифікації подій, а також автоматичного реагування за заздалегідь визначеними сценаріями. LibreNMS здатен автоматично визначати статус події (попередження,

критичні, інформаційні), ініціювати автоматичні дії, наприклад, запуск скриптів, формування звітів чи надсилання повідомлень. Інтеграція з Syslog дозволяє детально аналізувати журнали пристроїв, забезпечуючи швидке виявлення та локалізацію проблем у мережевій інфраструктурі.

2.2 Інтеграція LibreNMS з протоколами SNMP, Syslog, ICMP та іншими засобами моніторингу

Інтеграція LibreNMS з SNMP (Simple Network Management Protocol) є фундаментальною складовою системи, що забезпечує ефективний збір та аналіз метрик з мережевих пристроїв. LibreNMS підтримує всі основні версії SNMP (v1, v2c, v3), завдяки чому забезпечується сумісність з широким спектром обладнання, включаючи комутатори, маршрутизатори, сервери, точки доступу та інші пристрої [52].

Процес інтеграції базується на налаштуванні SNMP-спільнот або автентифікації через SNMP v3 з використанням шифрування для забезпечення безпеки передачі даних. LibreNMS автоматично здійснює SNMP-запити до пристроїв для отримання детальних метрик: завантаження процесора, використання оперативної пам’яті, температури, статусів портів та інтерфейсів, статистики трафіку, інформації про апаратну та програмну конфігурацію.

LibreNMS використовує стандартні та кастомні Management Information Bases (MIB) для максимальної сумісності з обладнанням різних виробників, що дозволяє автоматично ідентифікувати тип пристрою, збирати релевантні метрики та створювати спеціалізовані шаблони моніторингу [53].

Періодичне опитування SNMP забезпечує не лише збір метрик у реальному часі, але й історичний аналіз, завдяки чому легко виявляти тенденції, аномалії та проблеми продуктивності. LibreNMS також підтримує отримання SNMP trap-повідомлень, які дозволяють миттєво реагувати на критичні події без додаткових затримок на періодичне опитування.

Таким чином, глибока інтеграція LibreNMS з SNMP забезпечує ефективний та точний моніторинг мережевої інфраструктури, сприяє швидкому виявленню та вирішенню проблем, і значно спрощує адміністрування пристроїв завдяки автоматизації процесів збору та аналізу даних.

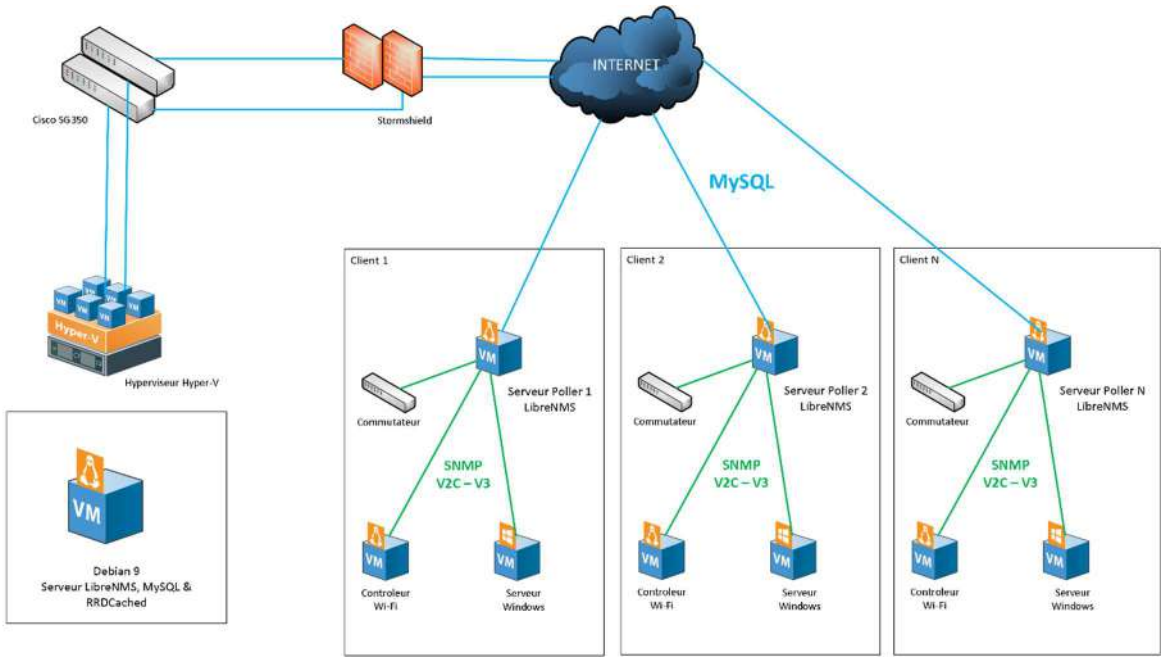


Рисунок 2.3 – Інтеграція LibreNMS із пристроями через SNMP

Наведена на рис. 2.3 схема демонструє інтеграцію LibreNMS із пристроями через SNMP. Центральний poller отримує дані від SNMP-агентів на комутаторах, маршрутизаторах і серверах, зберігає метрики, топологію трубопроводів (LLDP/CDP/xDP) і налаштовані SNMP-traps у RRD і базі даних, забезпечуючи автоматичне оновлення мережевих карт і оповіщень.

Такий підхід дозволяє безперервно збирати інформацію про використання CPU, пам’яті, трафіку та стан інтерфейсів, а також оперативно реагувати на події завдяки обробці трапів у реальному часі, що значно підвищує точність і своєчасність моніторингу.

LibreNMS підтримує інтеграцію з протоколом Syslog для централізованого збору та аналізу журналів подій, що суттєво розширює можливості моніторингу та реагування на інциденти. Використання Syslog дозволяє системі отримувати в реальному часі повідомлення про події з мережевих пристроїв: маршрутизаторів, комутаторів, серверів, точок доступу Wi-Fi тощо. Це сприяє оперативному

виявленню і реагуванню на критичні події, помилки, несанкціонований доступ або інші аномалії у роботі обладнання.

В LibreNMS налаштування прийому Syslog-повідомлень відбувається через спеціалізований сервер (зазвичай rsyslog або syslog-ng), що передає повідомлення у систему. Отримані дані класифікуються та зберігаються у внутрішній базі даних, де вони стають доступними для аналізу через веб-інтерфейс. Користувачі можуть переглядати журнали в реальному часі, застосовувати фільтрацію за типами пристроїв, рівнями важливості (severity) або ключовими словами.

На базі отриманих через Syslog подій можна формувати тригери й автоматичні сповіщення, що дозволяє швидко реагувати на критичні ситуації, наприклад, збої в роботі пристроїв, спроби вторгнення або несанкціоновані зміни конфігурації.

Таким чином, інтеграція LibreNMS з Syslog забезпечує не лише централізований збір логів, але й ефективну систему контролю й реагування на події, що значно підвищує оперативність і якість моніторингу інфраструктури.

LibreNMS використовує протокол ICMP як один з базових механізмів перевірки доступності мережевих пристроїв. Основною формою такої перевірки є надсилання ICMP Echo-запитів (ping), які дозволяють визначити, чи відповідає пристрій у мережі, і скільки часу займає обмін пакетами. Це забезпечує швидкий та малоресурсний спосіб виявлення недоступних вузлів у мережевій інфраструктурі.

На відміну від SNMP або інших протоколів збору даних, ICMP не вимагає жодної конфігурації на стороні пристрою, за винятком того, що він має приймати і відповідати на Echo-запити. Це особливо зручно для моніторингу простих пристроїв або пристроїв з обмеженими можливостями керування, таких як принтери, IoT-пристрої або шлюзи без SNMP-підтримки.

У LibreNMS ICMP-моніторинг зазвичай використовується як перша перевірка доступності перед виконанням глибших SNMP-опитувань. Якщо пристрій не відповідає на ICMP-запити, система може негайно зафіксувати його як недоступний і запустити відповідний тригер оповіщення, не витрачаючи ресурси на спроби збору додаткових даних.

Таким чином, використання ICMP у LibreNMS забезпечує базову, але важливу функцію контролю життєздатності пристроїв і дозволяє оперативно виявляти проблеми з доступністю в мережі ще до того, як буде зафіксовано відсутність специфічних метрик або сервісів.

Інтеграція LibreNMS з NetFlow та sFlow дає змогу здійснювати детальний аналіз мережевого трафіку, що проходить через ключові вузли інфраструктури, такі як маршрутизатори та L3-комутатори. Ці протоколи дозволяють отримувати інформацію про кожен мережевий потік: IP-адреси джерела й призначення, порти, обсяг переданих даних, тривалість сесій, тип трафіку та інші параметри, що дозволяє виявляти перевантаження, нетипову активність, потенційні атаки та порушення політик доступу.

LibreNMS самостійно не обробляє потоки NetFlow/sFlow, але підтримує інтеграцію з такими зовнішніми інструментами, як nfsen (NetFlow Sensor) або pmacct, які приймають трафік з пристроїв, що підтримують ці протоколи. У межах цієї інтеграції LibreNMS може відображати агреговані статистичні дані та надавати графічний інтерфейс для візуалізації обсягів трафіку за протоколами, пристроями або напрямками.

Це особливо корисно для виявлення “тяжких” користувачів, ненормального зростання трафіку, спроб сканування, витоків даних або DDoS-атак. Використання NetFlow або sFlow забезпечує глибше розуміння структури трафіку в мережі й дозволяє поєднати пасивний моніторинг (на основі метрик пристроїв) з активною аналітикою на рівні потоків, що значно розширює функціональні можливості LibreNMS у контексті безпеки та продуктивності мережі.

Інтеграція не є складною – достатньо налаштувати маршрутизатор на надсилання NetFlow/sFlow до обраного колектора, пов’язати його з LibreNMS через відповідний плагін або зовнішній модуль, і система почне приймати та обробляти інформацію про мережеві потоки в реальному часі.

2.3 Можливості LibreNMS у виявленні аномалій і подій безпеки

LibreNMS має розвинені механізми аналізу метрик у динаміці, що дозволяє виявляти відхилення від нормальної роботи пристроїв, які можуть свідчити про збої, перевантаження або потенційні інциденти безпеки. Аномалії виявляються шляхом постійного збору даних про продуктивність мережевих компонентів: інтерфейсів, процесорів, пам'яті, систем охолодження, дисків, а також параметрів, специфічних для певних типів пристроїв. Наприклад, у комутаторів можуть моніторитись STP-події, у точок доступу – кількість підключень, у серверів – навантаження на диски та процеси. LibreNMS дозволяє налаштовувати як абсолютні порогові значення (наприклад, CPU > 80% протягом 5 хвилин), так і логічні умови на комбінації кількох параметрів.

Система веде історію змін для кожного об'єкта моніторингу, що дозволяє не лише реагувати на одиничні сплески активності, а й проводити трендовий аналіз для виявлення повільних змін, таких як деградація продуктивності або поступове збільшення навантаження. Наприклад, поступове зростання трафіку на певному інтерфейсі протягом кількох тижнів може бути індикатором несанкціонованої активності або неефективного розподілу ресурсів.

У LibreNMS є можливість створювати спеціальні сповіщення на основі шаблонів, що дозволяє автоматично виявляти відхилення від профілю поведінки пристрою. Наприклад, для інтерфейсу можна задати умову, що якщо середній вхідний трафік перевищує середнє значення за попередні 24 години більше ніж на 300%, це є причиною для генерації попередження. Такий підхід дозволяє виявляти не лише критичні, а й латентні проблеми, які без аналізу в динаміці залишилися б непоміченими.

Також у LibreNMS підтримується візуалізація аномалій через графіки, де зміни у значеннях миттєво помітні завдяки кольоровому кодуванню або різким пікам. У поєднанні з оповіщеннями, історичними звітами та можливістю інтеграції з іншими системами безпеки, це робить LibreNMS дієвим інструментом не лише для моніторингу доступності, але і для проактивного виявлення аномалій, які можуть свідчити про інциденти або приховані загрози в мережевій інфраструктурі.

LibreNMS здійснює фіксацію та аналіз подій у мережі за допомогою кількох джерел, зокрема Syslog, SNMP traps та ICMP. Ці механізми дозволяють системі отримувати дані не лише внаслідок періодичного опитування пристроїв, а й у реальному часі – від самого обладнання у вигляді повідомлень про події.

Інтеграція з Syslog дає змогу централізовано приймати журнали подій від усіх пристроїв, які підтримують передачу логів. Такі повідомлення можуть містити інформацію про збої в інтерфейсах, перезавантаження, спроби доступу, зміни конфігурації, сигнали захисту від атак тощо. LibreNMS, працюючи спільно з syslog-сервером (наприклад, rsyslog або syslog-ng), приймає ці повідомлення, зберігає їх у базі даних і відображає в інтерфейсі з можливістю пошуку, фільтрації за пристроєм або важливістю. Це забезпечує адміністратора повною картиною подій у мережі.

SNMP traps – це інший важливий канал повідомлень, де пристрої самостійно ініціюють надсилання повідомлення до LibreNMS при настанні певної події, наприклад, зникнення живлення, відключення порту, перегрівання або помилка сервісу. Важливо те, що ці повідомлення надходять негайно, минаючи циклічний polling, що дозволяє скоротити час реакції системи на інцидент. LibreNMS автоматично фіксує ці події, відображає їх у системному журналі та, за потреби, генерує тригери на оповіщення.

ICMP використовується для регулярної перевірки досяжності пристроїв. Якщо хост не відповідає на ping, LibreNMS реєструє це як подію недоступності та створює інцидент. Це дозволяє швидко виявити вихід пристрою з ладу або проблеми з маршрутизацією.

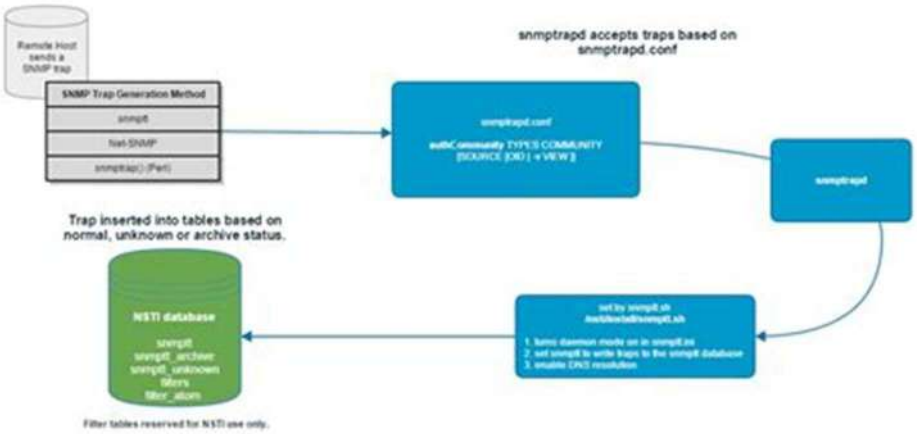


Рисунок 2.4 – Потік подій із пристроїв до LibreNMS.

Комбінування цих джерел – Syslog, SNMP traps і ICMP – забезпечує LibreNMS змогу фіксувати не лише зміни в продуктивності, а й ключові події, що відбуваються у системі в реальному часі. Такий підхід значно підвищує якість діагностики, скорочує час реагування на інциденти та дає змогу адміністраторам ефективно розслідувати проблеми або порушення в роботі мережі.

На схемі (Рис. 2.4) зображено потік подій із пристроїв до LibreNMS за допомогою SNMP traps, Syslog і ICMP перевірок. Пристрої надсилають SNMP traps на snmptrapd, який обробляє сигнали і передає їх в LibreNMS для занісення в журнал подій та активації тригерів. Syslog-сервер (rsyslog/syslog-ng) приймає повідомлення, передає їх у syslog.php LibreNMS, де вони класифікуються і також додаються до логів. ICMP пінги через fping перевіряють досяжність, і у випадку недоступності формують відповідні інциденти. Таке об’єднання джерел подій дозволяє LibreNMS оперативно фіксувати, класифікувати й реагувати на порушення в мережі, поєднуючи асинхронні повідомлення з активними перевірками пристроїв.

Система оповіщень LibreNMS є ключовим інструментом для оперативного реагування на відхилення від нормального стану мережевої інфраструктури. Вона дозволяє автоматично генерувати повідомлення про події, які відповідають заданим умовам, і надсилати їх відповідальним особам через зручні канали зв'язку. Оповіщення формуються на основі шаблонів або кастомізованих умов, що ґрунтуються на значеннях метрик, подіях з Syslog або SNMP traps, а також на втраті доступності пристроїв.

LibreNMS дозволяє гнучко налаштовувати умови спрацювання сповіщень, визначаючи порогові значення для будь-яких моніторингових параметрів – наприклад, завантаження CPU понад 85% протягом 5 хвилин, недоступність вузла за ICMP понад 3 цикли, або події типу «link down» на критичних інтерфейсах. Це дозволяє адаптувати систему під особливості конкретної мережі та уникнути хибних позитивів. Для кожного повідомлення можна налаштувати інтервал

повторної відправки, умови ескалації та автоматичне закриття інциденту після відновлення нормального стану.

Оповіщення можуть надсилатися різними каналами, включаючи електронну пошту, Telegram, Slack, Discord, Microsoft Teams, SMS, Webhooks, а також через кастомні скрипти. Це дозволяє легко інтегрувати LibreNMS з іншими системами моніторингу, керування інцидентами (наприклад, Zabbix, Grafana, PagerDuty) або автоматизованими платформами реагування.

Інтерфейс управління сповіщеннями в LibreNMS дозволяє переглядати історію повідомлень, їх статус (спрацьоване/відновлене), тип події та пов'язаний пристрій. Це забезпечує адміністраторам повний контроль над реагуванням системи на інциденти. У разі необхідності можна відкласти або тимчасово вимкнути окремі правила оповіщень, що є зручним у період технічного обслуговування.

Таким чином, система оповіщень LibreNMS реалізує не лише пасивне фіксування подій, а й активну стратегію реагування, що дозволяє скоротити час між виявленням проблеми та її усуненням, підвищуючи загальну надійність і безпеку мережі.

Інтеграція LibreNMS із зовнішніми сервісами відіграє важливу роль у підвищенні ефективності контролю безпеки та реагування на інциденти в мережевій інфраструктурі. Через відкритий REST API, підтримку Webhooks і плагінів, LibreNMS може передавати критичну інформацію в інші системи моніторингу, SIEM-рішення, системи керування інцидентами або хмарні сервіси аналітики.

Наприклад, завдяки Webhook-інтеграції система може негайно передати сповіщення про критичну подію до зовнішньої платформи, як-от Splunk, ELK Stack, Graylog або Microsoft Sentinel, для глибшого аналізу, збереження і кореляції з іншими подіями. Це дозволяє не лише бачити факт події в LibreNMS, а й співвіднести його з іншими сигналами безпеки, такими як логіни користувачів, підозрілі трафіки або аномальні дії на хостах.

Крім того, інтеграція з сервісами на кшталт PagerDuty або Opsgenie дає змогу побудувати систему ескалацій, коли критичні інциденти автоматично

перенаправляються до відповідальної чергової групи, скорочуючи час реагування на порушення. А підключення до месенджерів (Telegram, Slack, Discord) забезпечує миттєву доставку сповіщень, що важливо для оперативного інформування технічних команд.

LibreNMS також може передавати дані до систем графічної аналітики, таких як Grafana або Kibana, для побудови глибших інформаційних панелей і візуалізації трендів з безпеки: обсяги неуспішних підключень, зміни конфігурацій, частота падінь вузлів тощо.

Таким чином, інтеграція LibreNMS з іншими інструментами безпеки, логування й аналітики дозволяє вивести систему моніторингу за межі локального контролю окремих пристроїв і створити єдину екосистему для централізованого виявлення, кореляції та реагування на події, що значно підвищує загальний рівень інформаційної безпеки мережі.

2.4 Порівняння LibreNMS з альтернативами

LibreNMS вирізняється серед систем моніторингу Zabbix, Nagios та PRTG своєю автоматизацією, простотою налаштування та високим рівнем сумісності з сучасним мережевим обладнанням. На відміну від Nagios, який має статичну конфігурацію та потребує ручного додавання кожного вузла, LibreNMS забезпечує повністю автоматичне виявлення пристроїв (auto-discovery) та інтерфейсів завдяки використанню SNMP, LLDP/CDP, ARP, ICMP і скануванню підмереж. Це дозволяє швидко запускати систему навіть у великій інфраструктурі без значних витрат на ручну конфігурацію.

На відміну від Zabbix, який є потужною, але складнішою системою, LibreNMS має більш легкий процес розгортання та інтуїтивно зрозумілий інтерфейс користувача, орієнтований насамперед на мережевих адміністраторів. Він не потребує окремої конфігурації шаблонів для кожного пристрою – підтримка широкого спектра MIB реалізована вже в дистрибутиві, що робить моніторинг обладнання різних виробників доступним "із коробки". У той час як Zabbix

потребує більше часу на розгортання складних сценаріїв, LibreNMS дозволяє одразу почати роботу без потреби глибокого налаштування.

PRTG, хоч і має зручний інтерфейс та готові датчики, є переважно комерційним продуктом з ліцензійними обмеженнями. LibreNMS натомість є повністю відкритим і безкоштовним, із прозорою структурою та підтримкою спільноти. Крім того, LibreNMS має потужну систему REST API, що дозволяє інтегрувати його з зовнішніми сервісами, зокрема SIEM-системами, графічними візуалізаторами типу Grafana, системами авторизації через LDAP або Active Directory.

Ще однією сильною стороною LibreNMS є підтримка розподіленого моніторингу – distributed poller, який дозволяє масштабувати систему горизонтально. Nagios має обмежені можливості масштабування без використання зовнішніх інструментів, Zabbix потребує окремого налаштування проксі, а PRTG обмежений кількістю сенсорів без додаткових витрат. У LibreNMS розподілення опитування відбувається просто, без складної конфігурації, із підтримкою синхронізації через Memcached, Redis або MySQL-блокування.

У сфері безпеки LibreNMS має вбудовані можливості авторизації, розмежування доступу, логування активностей, підтримку HTTPS та інтеграцію з механізмами централізованої автентифікації. Це забезпечує контрольований доступ до системи і дозволяє адаптувати її до вимог інформаційної безпеки установи.

Таким чином, LibreNMS вигідно вирізняється поєднанням простоти, автоматизації, масштабованості та безкоштовної відкритої ліцензії, що робить його оптимальним вибором у випадках, коли потрібне швидке розгортання повноцінного моніторингу мережевої інфраструктури без складної конфігурації і вит LibreNMS демонструє високу масштабованість і гнучкість завдяки модульній архітектурі, що дозволяє адаптувати систему до будь-яких умов – від невеликих локальних мереж до масштабних розподілених інфраструктур із тисячами пристроїв. Основною перевагою є підтримка розподілених poller-ів, які можуть бути розміщені на окремих серверах і працювати паралельно, здійснюючи незалежне опитування пристроїв і зберігаючи результати у спільній базі. Це

забезпечує баланс навантаження без складної конфігурації, що відрізняє LibreNMS від, наприклад, Zabbix, де подібне масштабування потребує окремого проксі-сервера і значного втручання в конфігурацію.

LibreNMS легко адаптується до різних топологій і моделей адміністрування. Завдяки автоматичному виявленню пристроїв та інтерфейсів адміністратор може розгортати систему поетапно, не витрачаючи час на ручне додавання кожного вузла. Це вигідно вирізняє її від Nagios, де всі вузли описуються вручну в конфігураційних файлах, або навіть PRTG, який хоча й має авто-дискавері, але обмежений безкоштовною кількістю сенсорів і менш гнучкий у модифікації шаблонів моніторингу.

Процес встановлення LibreNMS є доволі простим: він має детальну офіційну документацію, підтримує автоматизацію розгортання через Ansible, Docker або скрипти, й може бути розгорнутий на будь-якій сучасній системі Linux. Усі налаштування відбуваються через веб-інтерфейс або REST API, без необхідності вручну редагувати конфігураційні файли.

Гнучкість LibreNMS також проявляється у підтримці великої кількості протоколів, зовнішніх плагінів, а також можливості писати власні модулі для обробки нестандартних пристроїв. Це дає змогу адаптувати систему під специфічні потреби – наприклад, моніторити обладнання освітньої установи, промислові контролери або IoT-пристрої, що не завжди можливо у Zabbix чи Nagios без значної кастомізації.

Таким чином, у порівнянні з іншими популярними системами моніторингу LibreNMS демонструє оптимальне поєднання масштабованості, гнучкості та простоти у розгортанні, що робить його особливо привабливим для установ, які прагнуть отримати функціональний та адаптивний інструмент без зайвих витрат і складності конфігурації.рат.

LibreNMS пропонує широкий спектр можливостей у сфері підтримки безпеки, сучасних протоколів та інтеграцій, що вигідно відрізняє його від інших систем моніторингу. Однією з основ безпечної роботи є повна підтримка HTTPS для захищеного доступу до веб-інтерфейсу, що дозволяє захистити облікові дані та передану інформацію від перехоплення. Додатково, система дозволяє інтеграцію з

LDAP та Active Directory, що забезпечує централізоване керування автентифікацією користувачів і впровадження ролей з обмеженням доступу на основі привілеїв. Це особливо важливо для установ з кількома рівнями відповідальності або зонованим доступом до мережевої інфраструктури.

LibreNMS підтримує основні мережеві протоколи для збору даних: SNMP v1/v2c/v3, ICMP (ping), LLDP/CDP, ARP, Syslog, SNMP traps, NetFlow, sFlow, IPMI, а також API-інтерфейси для взаємодії з хмарними або кастомними пристроями. На відміну від Nagios, де більшість таких інтеграцій реалізується через зовнішні плагіни, у LibreNMS більшість можливостей вже вбудована або доступна через модулі, що спрощує налаштування й зменшує ризик конфігураційних помилок.

З точки зору інтеграцій, LibreNMS має відкритий REST API, який дозволяє читати, створювати та оновлювати інформацію про пристрої, сповіщення, події та інші об’єкти системи. Це відкриває широкі можливості для інтеграції з зовнішніми платформами: SIEM-системами (Splunk, Graylog), сервісами реагування на інциденти (PagerDuty, Opsgenie), візуалізаційними інструментами (Grafana, Kibana), CMDB та хмарними моніторинговими рішеннями. Також підтримуються Webhooks для реакції на події в реальному часі – наприклад, надсилання даних до Telegram, Slack або Microsoft Teams.

Важливою особливістю є наявність системи контролю змін конфігурацій пристроїв через інтеграцію з Oxidized, що дозволяє зберігати версії конфігурацій маршрутизаторів і комутаторів, аналізувати різниці між ними та виявляти несанкціоновані зміни. Це додає ще один рівень безпеки та контролю над мережевою інфраструктурою.

Отже, LibreNMS забезпечує комплексну підтримку сучасних протоколів, безпечний контроль доступу та гнучкі механізми інтеграції з іншими системами, що в сукупності дозволяє йому бути не лише засобом моніторингу, а й важливою ланкою в системі мережевої безпеки організації.

2.5 Висновки до розділу

У ході аналізу функціональних можливостей LibreNMS, його архітектури, технологічної сумісності та переваг над конкуруючими рішеннями було встановлено, що дана система є потужним, гнучким і ефективним інструментом для реалізації комплексного моніторингу мережевої інфраструктури з орієнтацією на безпеку, масштабованість та автоматизацію. Система LibreNMS побудована за клієнт-серверною архітектурою з чітким поділом функціональних компонентів: веб-інтерфейсу, бази даних, poller-ів і системи зберігання часових рядів, що дозволяє легко масштабувати інсталяцію відповідно до розміру мережі та потреб замовника.

Використання LibreNMS як системи моніторингу забезпечує повний цикл управління: від автоматичного виявлення пристроїв і збору метрик до обробки подій, генерації сповіщень і інтеграції з іншими засобами інформаційної безпеки. Особливо вагомою перевагою LibreNMS є підтримка великої кількості технологій і протоколів, таких як SNMP, ICMP, Syslog, NetFlow/sFlow, LLDP/CDP, що дозволяє інтегрувати практично будь-яке сучасне мережеве обладнання без необхідності створення кастомних скриптів чи шаблонів. Уся архітектура орієнтована на відкритість та адаптивність: REST API, webhook-и, підтримка зовнішніх колекторів, можливість зберігання та порівняння конфігурацій, а також використання баз даних RRD, MariaDB і Redis роблять систему конкурентоспроможною у сфері інфраструктурного моніторингу навіть порівняно з комерційними рішеннями.

Серед особливостей, які виокремлюють LibreNMS на тлі альтернатив, слід зазначити інтуїтивно зрозумілий інтерфейс, просте налаштування, підтримку auto-discovery, а також масштабованість завдяки distributed polling. У порівнянні з Zabbix чи Nagios, які вимагають значного часу на конфігурацію та оновлення шаблонів, LibreNMS дозволяє скоротити час розгортання до мінімуму, при цьому не поступаючись у можливостях виявлення, аналізу й реагування. У разі порівняння з PRTG, LibreNMS має незаперечну перевагу в тому, що не має

ліцензійних обмежень, повністю відкритий, а також не потребує додаткових витрат на масштабування функціоналу.

Завдяки підтримці централізованого збору подій із Syslog, SNMP traps та ICMP, система дозволяє гнучко аналізувати події, класифікувати їх за рівнем важливості, застосовувати політики реагування, створювати шаблони для сповіщень і керувати інцидентами в реальному часі. Інтеграція з зовнішніми платформами безпеки (SIEM, аналітика, Telegram, Microsoft Teams, Grafana, Kibana) виводить LibreNMS за межі звичайного моніторингу до ролі інтегрованого аналітичного інструмента в екосистемі організації.

З погляду безпеки, LibreNMS підтримує HTTPS, аутентифікацію через LDAP/AD, обмеження доступу на основі ролей, що дозволяє легко впровадити його в середовищах з високими вимогами до контролю доступу й захисту конфіденційної інформації. Також важливою перевагою є здатність системи зберігати історію змін, вести аудит і будувати гнучку політику сповіщень, що сприяє швидкому виявленню інцидентів, їх локалізації та усуненню.

Усе вищезазначене дозволяє стверджувати, що LibreNMS є не лише зручним, але й стратегічно ефективним рішенням для побудови системи моніторингу безпеки у сучасному мережевому середовищі. Його гнучкість, розширюваність, підтримка великого спектра стандартів і відкритість є ключовими перевагами, які роблять цю систему однією з найкращих опцій у своєму класі. У контексті шкільної або освітньої мережі LibreNMS дозволяє реалізувати контроль за пристроями, точками доступу, відеонаглядом і сервісами з мінімальними зусиллями, підтримуючи масштабування інфраструктури в майбутньому.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ МОНІТОРИНГУ БЕЗПЕКИ НА БАЗІ LIBRENMS

3.1 Середовище розгортання

У запропонованій реалізації система моніторингу LibreNMS розгортається у вигляді окремої віртуальної машини (ВМ) на гіпервізорі Hyper-V, що є стандартною платформою в середовищах з інфраструктурою Microsoft. Hyper-V встановлений на фізичному сервері, який виконує роль хоста віртуалізації та забезпечує ізольоване середовище для функціонування окремих сервісів. Така архітектура дозволяє логічно відокремити функції моніторингу від інших компонентів IT-інфраструктури, а також спрощує резервне копіювання, масштабування й адміністрування системи.

Фізичний сервер має встановлену 64-бітну серверну операційну систему Windows Server 2022, яка виступає в ролі основної платформи для Hyper-V. Завдяки підтримці апаратної віртуалізації Intel VT-x, гіпервізор надає повний набір інструментів для створення, керування та моніторингу віртуальних машин, включаючи налаштування мережевих інтерфейсів, дисків, резервного копіювання та ізоляції ресурсів.

Віртуальна машина, на якій розгортається LibreNMS, створюється з виділенням окремих обчислювальних ресурсів: CPU, оперативної пам’яті, дискового простору та віртуального мережевого інтерфейсу. На цю ВМ встановлюється Ubuntu Server 22.04 LTS як базова операційна система з подальшим встановленням усіх необхідних компонентів LibreNMS. Розгортання у віртуальному середовищі дозволяє за потреби легко змінювати конфігурацію системи: збільшувати ресурси, клонувати ВМ, створювати контрольні точки (checkpoints) для тестування або оновлення, а також розгорнути резервну копію у разі збоїв.

Обраний підхід також полегшує масштабування системи в майбутньому: можливо підключити додаткові poller-и у вигляді окремих ВМ або контейнерів, забезпечуючи горизонтальне розширення моніторингової інфраструктури без прив’язки до фізичної машини. Застосування віртуалізації дозволяє оптимізувати

використання апаратних ресурсів, підвищити відмовостійкість та ізолювати моніторингове середовище від основної мережевої інфраструктури з точки зору безпеки.

В якості базової платформи для розгортання системи моніторингу LibreNMS було використано фізичний сервер HP ProLiant DL330 G6, що забезпечив належний рівень ресурсів для створення повноцінного віртуального середовища. Роль гіпервізора виконує Hyper-V, встановлений на серверній операційній системі Windows Server. У межах цієї інфраструктури було створено окрему віртуальну машину, що стала цільовим середовищем для розміщення LibreNMS. Обрана ВМ отримала виділені апаратні ресурси, що відповідають як поточним потребам, так і можливості подальшого масштабування без міграції чи зміни архітектури.

На віртуальну машину було призначено чотириядерний віртуальний процесор, який базується на фізичних ядрах центрального процесора Xeon. Це забезпечило необхідну обчислювальну потужність для опитування великої кількості пристроїв одночасно, обробки трафіку SNMP, збереження даних до бази, генерації графіків і виконання аналітики на рівні метрик у реальному часі. За обсягом оперативної пам’яті було виділено 16 ГБ, що дозволило розмістити в оперативній пам’яті ключові служби: веб-сервер, PHP-процеси, MariaDB, Redis, RRDcached, а також залишити достатній буфер під роботу системи кешування та запобігання надмірному навантаженню на файлову підсистему.

Щодо дискового простору, то під систему було зарезервовано віртуальний диск об’ємом 150 ГБ з розміщенням на фізичному SSD-масиві, що працює через контролер з підтримкою RAID. Низька латентність та висока пропускна здатність SSD накопичувачів забезпечили необхідну продуктивність для інтенсивного запису RRD-файлів, роботи бази даних, а також доступу до логів і файлів журналів. У подальшому була реалізована можливість створення окремого логічного тому для резервного копіювання та довготривалого зберігання журналів. З огляду на те, що сервер має шість SSD-дисків по одному терабайту, обраний підхід не створив загрози дефіциту ресурсів навіть при збільшенні навантаження.

Окремо було враховано мережеву складову. У конфігурації сервера існувала виділена мережева карта, зарезервована виключно під потреби моніторингової системи. Віртуальну машину було прив’язано до цього фізичного адаптера через віртуальний комутатор Hyper-V, що забезпечило ізольований та гарантовано доступний канал зв’язку. Мережевий інтерфейс функціонує на швидкості 1 Gbit/s, що є достатнім для опитування кількох сотень пристроїв, надсилання запитів SNMP, прийому Syslog-повідомлень, обробки ICMP-перевірок та обміну даними з іншими службами або інтеграційними системами. Це також дозволило уникнути затримок і перевантаження спільного мережевого каналу, що могло б вплинути на точність фіксації подій або вчасність сповіщень.

Таким чином, виділені ресурси віртуального середовища відповідали як мінімальним, так і рекомендованим параметрам для продуктивної та стабільної роботи LibreNMS. Уся конфігурація була розрахована на тривале безперервне функціонування в режимі 24/7 з урахуванням резерву для майбутнього розширення, можливості горизонтального масштабування через розподілених poller-ів, а також з дотриманням вимог щодо швидкодії, надійності та ізоляції трафіку моніторингу в межах локального середовища.

У якості базової серверної операційної системи для віртуальної машини, на якій розгорнуто систему моніторингу LibreNMS, було обрано дистрибутив Ubuntu Server версії 22.04 LTS (Long Term Support). Це рішення ґрунтується на низці технічних, практичних та стратегічних міркувань, які охоплюють питання стабільності, підтримки, сумісності з необхідним програмним забезпеченням, зручності адміністрування та відповідності рекомендаціям розробників LibreNMS.

Ubuntu Server 22.04 LTS є офіційно підтримуваною версією з тривалим циклом оновлень безпеки, який триватиме щонайменше до 2027 року. Це забезпечує стабільну платформу для безперервної експлуатації без ризику втрати підтримки в найближчі роки. У порівнянні з проміжними або нестабільними випусками, версія LTS має ретельно протестовані компоненти ядра, стабільні версії бібліотек і сервісів, що особливо важливо для серверних систем, які функціонують у цілодобовому режимі.

LibreNMS офіційно рекомендує Ubuntu Server як одну з основних платформ для розгортання, а всі інструкції з встановлення та оновлення мають повну сумісність саме з цим дистрибутивом. На відміну від більш складних або менш популярних систем (наприклад, Arch Linux або CentOS Stream), Ubuntu забезпечує найкращу інтеграцію з менеджером пакетів APT, дозволяє швидко інсталювати потрібні залежності, а також підтримує офіційні PPA-репозиторії для оновлення компонентів PHP, MariaDB, Redis, SNMP та ін.

Окрім цього, Ubuntu має високу популярність у спільнотах адміністраторів і DevOps-інженерів, що забезпечує наявність великої кількості якісної документації, прикладів конфігурацій, відповідей на питання й шаблонів автоматизації, зокрема у вигляді Ansible-ролей та Docker-образів. Це значно спрощує обслуговування, відновлення після збоїв, оновлення компонентів та розширення функціоналу без потреби у глибокому переписуванні сценаріїв або адаптації до нестандартних ОС.

Ubuntu Server також чудово оптимізований для роботи у віртуалізованих середовищах, включно з Hyper-V. Він підтримує встановлення через cloud-init, забезпечує коректну інтеграцію з драйверами Hyper-V, підтримує динамічне розширення ресурсів і працює без додаткових налаштувань із віртуальними мережевими адаптерами та сховищами. Це дозволило швидко налаштувати систему без необхідності встановлення сторонніх модулів ядра або обхідних рішень.

З погляду безпеки, Ubuntu дозволив впровадити обмеження доступу через SSH, використати UFW для контролю портів, а також інтегрувати TLS-сертифікати для захищеного доступу до веб-інтерфейсу LibreNMS. Всі сервіси – Apache або Nginx, MariaDB, PHP, SNMPD – повністю підтримуються в межах стандартних репозиторіїв, що спрощує їх обслуговування та оновлення.

Таким чином, використання Ubuntu Server 22.04 LTS як базової платформи дало змогу реалізувати надійне, передбачуване та масштабоване середовище для впровадження LibreNMS, яке відповідає сучасним вимогам до серверної інфраструктури, безпеки та довгострокової підтримки.

Для повноцінного функціонування системи моніторингу LibreNMS у середовищі Ubuntu Server 22.04 LTS було встановлено та налаштовано низку

обов’язкових програмних компонентів, які забезпечують взаємодію всіх підсистем, збір метрик, зберігання даних, обробку подій та надання доступу через веб-інтерфейс. Усі необхідні залежності були інстальовані через офіційні репозиторії Ubuntu або через рекомендовані сторонні джерела з використанням менеджера пакетів APT. Комплексна взаємодія між цими компонентами дозволила реалізувати стабільне серверне середовище з підтримкою автоматичного моніторингу, сповіщень, візуалізації та інтеграції з зовнішніми системами.

В якості веб-сервера було обрано Apache2, який забезпечив стабільну роботу PHP-додатку LibreNMS та підтримку HTTPS через модуль SSL. У процесі налаштування було активовано необхідні модулі, зокрема `mod_rewrite`, `mod_php` та `mod_ssl`. Apache відзначився як сумісне й надійне рішення, яке повністю підтримується офіційною документацією LibreNMS і не потребує додаткової адаптації при оновленнях.

Для зберігання даних LibreNMS використовує реляційну СУБД MariaDB, що була інстальована та налаштована відповідно до вимог – з використанням InnoDB, підтримкою таблиць великих обсягів та ввімкненою підтримкою utf8mb4. Було здійснено тонке налаштування параметрів конфігурації (зокрема `innodb_buffer_pool_size`, `max_connections`, `query_cache_size`), що дозволило оптимізувати продуктивність у середовищі з великою кількістю активних пристроїв. Альтернативно могла бути використана MySQL, однак MariaDB продемонструвала кращу гнучкість у розгортанні на Ubuntu.

Критичною залежністю є також PHP – LibreNMS вимагає використання актуальної версії з підтримкою широкого набору розширень. У середовищі було встановлено PHP 8.1 з модулями `php-mysql`, `php-cli`, `php-gd`, `php-snmp`, `php-curl`, `php-mbstring`, `php-xml`, `php-zip`, `php-bcmath` та `php-tokenizer`. Це забезпечило повну сумісність із вимогами системи та стабільну роботу як командного інтерфейсу, так і веб-доступу.

Для збору метрик по SNMP-протоколу було встановлено демон `snmpd` із відповідною конфігурацією community-рядків та дозволів доступу. LibreNMS виступає в ролі SNMP-клієнта, що регулярно опитує пристрої мережі, тож стабільна робота SNMP-серверу на стороні ОС дозволила забезпечити базу для

інтеграції з комутаторами, маршрутизаторами, серверами, точками доступу та іншими вузлами інфраструктури.

Для кешування та прискорення доступу до даних у LibreNMS було налаштовано Redis як основний механізм обробки тимчасових даних і зберігання проміжних результатів. У якості альтернативи могла бути використана служба Memcached, однак Redis виявився ефективнішим у сценаріях з великою кількістю одночасних запитів. Redis також використовувався у модулі обробки сповіщень, а також у процесі взаємодії з розподіленими poller-ами при масштабуванні системи.

Для реалізації регулярного запуску скриптів опитування пристроїв та оновлення бази графіків було використано `cron` через systemd. Завдяки вбудованим таймерам LibreNMS, які працюють через `cron.d`, система автоматично виконує регулярні завдання без необхідності ручного втручання. У зв'язці з цим також було встановлено утиліту `fping`, що дозволяє ефективно перевіряти досяжність пристроїв за допомогою ICMP без значного навантаження на систему.

З міркувань безпеки було налаштовано обмежений доступ до віртуальної машини LibreNMS через SSH. Було змінено стандартний порт, увімкнено автентифікацію за SSH-ключем, а також обмежено перелік дозволених користувачів. Це дозволило ізолювати систему від несанкціонованого доступу та підвищити загальний рівень захищеності середовища моніторингу.

Таким чином, перелік і конфігурація програмних компонентів, що були встановлені в Ubuntu Server, сформували повноцінну інфраструктуру для роботи LibreNMS. Усі залежності інтегрувалися стабільно, без конфліктів, а взаємодія між сервісами реалізована через стандартні системні механізми, що гарантує надійність роботи у довготривалій перспективі.

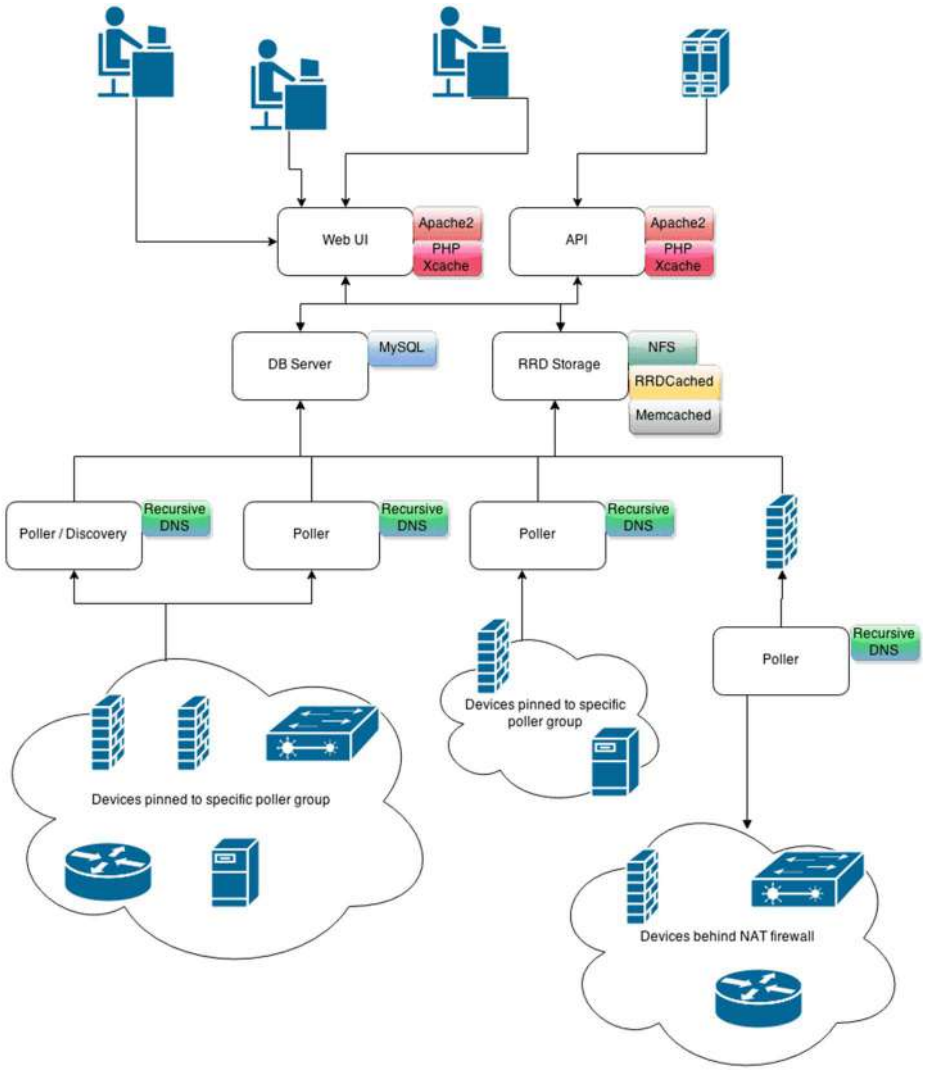


Рисунок 3.1 – Модульна архітектура LibreNMS

Схема на рис. 3.1 демонструє модульну архітектуру LibreNMS, що ілюструє взаємодію між ключовими компонентами системи: веб-інтерфейс (Web UI), REST API, базою даних (MySQL/MariaDB), сховищем RRD (RRDcached або NFS), кешем (Memcached/Redis) і розподіленими poller-ами (з discovery-модулями). На схемі видно детальну взаємозалежність між компонентами: web/UI надсилає запити до бази та кешу і отримує графіки з RRD, poller-и в реальному часі збирають метрики зі SNMP-агентів, оновлюють RRD і базу даних, а mutex-механізми забезпечують синхронний доступ до ресурсів. Ця архітектура підкреслює масштабованість системи, можливість горизонтального розширення через окремі poller-вузли та гарантує узгодженість даних завдяки механізмам синхронізації via Redis/Memcached.

3.2 Налаштування LibreNMS для моніторингу мережевого обладнання

У рамках налаштування системи LibreNMS після завершення встановлення всієї програмної інфраструктури було виконано початкову конфігурацію веб-інтерфейсу та підготовку до моніторингу мережевого обладнання. Після запуску веб-інтерфейсу LibreNMS через браузер було відкрито початковий майстер налаштування, який автоматично перевірив наявність необхідних залежностей, коректність налаштування прав доступу до директорій, зв'язок із базою даних та підтримку PHP-розширень. Всі попередні перевірки було пройдено успішно, після чого система дозволила перейти до створення адміністративного облікового запису.

Було створено обліковий запис адміністратора з унікальним іменем користувача, складним паролем та привілеями повного доступу до всіх функцій системи. Цей обліковий запис надалі використовувався для налаштування параметрів SNMP, мережевих діапазонів, політик сповіщень і прав користувачів.

Після входу в систему через веб-інтерфейс було здійснено перехід до розділу глобальних налаштувань, де було перевірено налаштування SNMP. Було активовано стандартну SNMP-community рядок з назвою `public` та обмежено доступ до SNMP-опитування лише з IP-адреси сервера LibreNMS, що підвищило рівень безпеки. У конфігураційному файлі `config.php` було вручну додано параметри SNMP, що дозволяють правильно інтерпретувати відповіді пристроїв Mikrotik.

Для перевірки коректності взаємодії SNMP між сервером LibreNMS та обладнанням Mikrotik було використано команду `snmpwalk`, яка дозволила переконатися у відповідях пристроїв за вказаним community та версією протоколу SNMP. Отримані відповіді включали OID і назви інтерфейсів, що підтвердило готовність пристроїв до моніторингу.

Після перевірки було виконано налаштування функції автоматичного виявлення обладнання за допомогою механізмів auto-discovery. У розділі «Add Device» було активовано модуль виявлення у вказаній підмережі 192.168.100.0/24, де знаходилися керовані комутатори. Було увімкнено протоколи розпізнавання

CDP, LLDP, а також ARP та SNMP discovery, що дозволило системі самостійно виявити наявні в мережі пристрої з активним SNMP-агентом.

У результаті автоматичного виявлення LibreNMS ідентифікував чотири комутатори Mikrotik CRS326-24G-2S+RM, які були зареєстровані в базі даних системи як окремі пристрої. Кожному з них автоматично було присвоєно ім'я відповідно до SNMP-hostname, визначено тип обладнання, операційну систему RouterOS, модель, кількість інтерфейсів, MAC-адреси та серійні номери. Також було автоматично зібрано інформацію про CPU, RAM, завантаження портів, IP-адреси, версію мікропрограмного забезпечення та аптайм.

Етапи налаштування системи моніторингу:

1. Відкрито веб-інтерфейс LibreNMS та завершено майстер первинного налаштування

2. Створено адміністративний обліковий запис із повними правами доступу

3. Перевірено підтримку SNMP і налаштовано community `public` на обладнанні та сервері

4. Виконано команду `snmpwalk` для перевірки відповідей пристроїв Mikrotik

5. Увімкнено auto-discovery в підмережі 192.168.100.0/24 із підтримкою LLDP, CDP, ARP, SNMP

6. Виявлено 4 пристрої Mikrotik CRS326-24G-2S+RM, які додано до системи моніторингу

7. Перевірено доступність пристроїв у списку, а також наявність базових метрик

Для кожного з пристроїв було збережено базовий конфігураційний знімок, після чого вони стали доступними для подальшого налаштування, групування, побудови графіків, створення сповіщень і інтеграції в загальну політику контролю за мережею.

Після початкового розгортання LibreNMS і успішного виявлення пристроїв було проведено оптимізацію параметрів опитування та оновлення метрик з метою забезпечення стабільного функціонування системи при мінімальному навантаженні на мережеву інфраструктуру. Основною метою цього етапу було

досягнення балансу між частотою збору даних, точністю моніторингу та економією системних ресурсів.

Для цього було змінено стандартний інтервал опитування пристроїв, який за замовчуванням становить 5 хвилин. У середовищі з високою щільністю пристроїв і постійним трафіком цей інтервал було залишено незмінним для критичних вузлів, таких як маршрутизатори та комутатори ядра, але для менш важливих пристроїв або кінцевих точок його збільшено до 10–15 хвилин через застосування політик у розділі «Device Settings». Це дозволило зменшити загальну кількість SNMP-запитів і уникнути перевантаження як мережі, так і poller-процесів.

Було активовано використання утиліти `fping` як основного інструменту для визначення доступності пристроїв за допомогою ICMP. На відміну від стандартної утиліти `ping`, `fping` дозволяє паралельно перевіряти велику кількість адрес, що зменшило час опитування при збереженні точності визначення відмов. Відповідно до документації LibreNMS, шлях до `fping` було прописано в конфігураційному файлі `config.php`, а також протестовано відповідність параметрів доступу до системи. Було встановлено мінімальний тайм-аут ICMP-запитів, що дозволило швидше виявляти недоступні вузли.

Щодо SNMP-тайм-аутів і кількості повторних спроб, було скориговано значення відповідно до типу обладнання. Для пристроїв Mikrotik тайм-аут залишено на рівні 1 секунди з двома повторними запитами, що дало змогу враховувати незначні затримки в обробці, зберігаючи при цьому загальну стабільність опитування. У пристроїв із повільнішим процесором або нестабільним з’єднанням SNMP-тайм-аут збільшено до 2 секунд.

Крім того, в налаштуваннях системи було обмежено одночасну кількість активних poller-процесів, що дозволило уникати пікового навантаження на CPU сервера в моменти опитування великої кількості вузлів. Було активовано логування часу виконання опитувань кожного пристрою, що дало змогу в подальшому проводити аналіз і автоматично коригувати пріоритети або інтервали відповідно до динаміки змін у мережевому середовищі.

У результаті проведених налаштувань система стала працювати стабільніше, із передбачуваним навантаженням і без втрати критичних даних. Метрики

оновлювалися в оптимальному режимі, забезпечуючи достатню деталізацію графіків і швидку реакцію в разі відмови чи перевищення порогових значень. Оптимізація була завершена шляхом спостереження за середнім часом опитування всіх пристроїв, що не перевищував 30 секунд при активних 50–60 мережевих вузлах.

Після налаштування базового моніторингу було здійснено впорядкування обладнання в LibreNMS через механізми логічного групування, категоризації та застосування тегів. Це дозволило сформувати чітку структуру мережевої інфраструктури в інтерфейсі системи, полегшити навігацію між пристроями, а також забезпечити можливість гнучкого фільтрування під час побудови дашбордів, графіків і сповіщень.

Було використано вбудовану функцію «Device Groups», що дозволила створити декілька логічних груп відповідно до топологічного розміщення та функціонального призначення обладнання. Зокрема, пристрої Mikrotik CRS326, які були розгорнуті в різних сегментах мережі (кореневий комутатор, розподільчі вузли, точки агрегації), були віднесені до окремих груп: "Core Switches", "Distribution Layer", "Edge Devices". Групи формувалися за умовами, що базувалися на IP-адресах, SNMP-моделях або іменах хостів, що значно спростило процес класифікації без необхідності ручного додавання кожного пристрою.

У паралель до цього в інтерфейсі LibreNMS було використано механізм «Device Types» для категоризації пристроїв за типом обладнання. Комутатори, маршрутизатори, точки доступу та сервери були автоматично класифіковані системою, але за потреби ця інформація була доповнена вручну для більшої точності. Було уточнено інформацію про виробника, модель та операційну систему, що у подальшому дозволило будувати узагальнені аналітичні звіти щодо стану обладнання певного типу чи певного вендора.

Для деталізації і кращої видимості кожному пристрою було присвоєно користувацькі теги. Теги були створені для вказівки на фізичну локацію (наприклад, ` corps-1`, `floor-2`, `server-room`), критичність (`core`, `last`), призначення (`video`, `wifi`, `control`). Теги давали змогу виконувати швидкий пошук, фільтрацію в таблицях пристроїв, а також використовувалися як умови для

сповіщень або автоматичного застосування шаблонів перевірок. Ці мітки дозволили реалізувати логічну багаторівневу класифікацію без дублювання ієрархій та зберегли гнучкість у подальшій зміні політик моніторингу.

Завдяки впровадженню згаданих механізмів навігація в інтерфейсі LibreNMS стала значно ефективнішою, зменшився час на пошук пристрою в разі інциденту, а побудова агрегованих графіків і логічних груп дозволила забезпечити централізований контроль за всіма вузлами критичної інфраструктури. Усі нові пристрої, що автоматично додаються через auto-discovery, одразу потрапляють до відповідних груп за заданими умовами, що виключає потребу ручної класифікації.

Після завершення додавання пристроїв до системи LibreNMS та оптимізації параметрів опитування було виконано перевірку достовірності отримуваних даних, їх візуалізацію у вигляді графіків та контроль стану пристроїв через інтерфейс журналів подій. Основною метою цього етапу було впевнитися в коректності збору метрик, відображенні динаміки навантаження на інтерфейси, достовірності логів, а також у працездатності механізмів фіксації змін у статусі пристроїв.

У розділі кожного окремого пристрою було відкрито вкладку «Health» та «Ports», де зчитувалися показники навантаження на інтерфейси, використання процесора, оперативної пам’яті, температури пристрою (за наявності датчиків), а також інші дані, що надходили через SNMP. Було перевірено, що всі основні графіки формуються коректно, зчитуються у реальному часі та оновлюються відповідно до заданого інтервалу опитування. На графіках можна було чітко спостерігати тенденції трафіку у вхідному та вихідному напрямках, зниження або підвищення активності інтерфейсів, а також тривалість роботи пристрою без перезавантаження.

Особливу увагу було приділено вкладці «Graphs», де доступні як агреговані, так і детальні графіки по кожному інтерфейсу пристрою. За допомогою порівняння фактичного навантаження на порти з очікуваними показниками було встановлено, що SNMP-запити обробляються правильно, дані надходять у повному обсязі, і RRD-файли формуються без збоїв. Здійснено перевірку графіків за різні часові інтервали – від останньої години до 24 годин, тижня і місяця – що дозволило переконатися у працездатності зберігання історичних метрик.

Після перевірки графічної візуалізації було також здійснено аналіз журналів подій у вкладках «Eventlog» і «Syslog», де фіксуються всі зміни стану пристроїв, виявлені порушення, повідомлення SNMP traps та повідомлення, надіслані через Syslog. У журналі подій автоматично зберігалася інформація про зміну статусу пристрою (up/down), втрату зв'язку, перевищення порогових значень трафіку, зміни у конфігурації SNMP-параметрів та інші інциденти. Окремо перевірено, що всі пристрої Mikrotik CRS326 надіслали базові лог-повідомлення через Syslog і SNMP traps, які успішно були зареєстровані LibreNMS.

Було здійснено контроль станів інтерфейсів – кожен порт, який було активовано на пристроях, мав індикатор статусу (up/down), статистику помилок, кількість отриманих та переданих байтів, а також графічне відображення активності за обраний проміжок часу. Усі інтерфейси з проблемами (наприклад, фізичне відключення, помилки передачі, низька пропускна здатність) були позначені відповідними іконками й автоматично виділені в загальному списку.

У результаті перевірки було підтверджено, що LibreNMS повністю забезпечує точний та безперервний збір мережевих метрик, їх візуалізацію та логування подій. Система дозволила не лише оперативно отримувати стан пристроїв у вигляді графіків і логів, а й формувати базу для подальшого аналізу навантаження, планування пропускної здатності та реагування на інциденти. Усі перевірені функції працювали у відповідності до документації та не виявили критичних відхилень у роботі.

3.3 Реалізація сповіщень, логування подій безпеки та візуалізація даних

У межах реалізації системи сповіщень у LibreNMS було повністю налаштовано механізми генерації, обробки та доставки алертів відповідно до подій, виявлених у процесі моніторингу мережевого обладнання. Основною метою конфігурації було забезпечити оперативне інформування адміністратора про критичні відхилення в роботі пристроїв, падіння інтерфейсів, втрату зв'язку або перевищення порогових значень за ключовими метриками.

У веб-інтерфейсі LibreNMS через розділ "Alert Rules" було створено базовий набір умов для спрацювання сповіщень. Серед них – втрата доступу до пристрою, зміна стану інтерфейсу на down, перевищення 80% завантаження каналу, а також порушення граничних значень температури або використання пам’яті. Кожне з правил було налаштовано з урахуванням специфіки обладнання Mikrotik CRS326 і їх топологічного положення, що дозволило уникнути надмірної генерації алертів у разі короткочасних або неважливих відхилень. Для кожного правила було встановлено рівень критичності – warning або critical – що надалі використовувалося для фільтрації і пріоритезації повідомлень.

У розділі "Alert Transport" було налаштовано механізм надсилання сповіщень через електронну пошту. Для цього використано внутрішній SMTP-сервер організації з підтримкою TLS-шифрування, що гарантувало захищену доставку повідомлень. Було вказано адресу відправника, порт, домен, логін та пароль для автентифікації, після чого система пройшла перевірку з надсиланням тестового листа на вказану адміністративну пошту. Повідомлення успішно надійшло, що підтвердило коректність параметрів SMTP-транспорту.

У вкладці шаблонів повідомлень було створено індивідуальний шаблон з урахуванням технічних вимог – включено динамічні змінні на кшталт імені пристрою, IP-адреси, опису інтерфейсу, часу інциденту, рівня критичності та гіперпосилання на відповідний запис у LibreNMS. Форматування виконано у вигляді таблиці HTML для кращої читабельності повідомлень. Тестування шаблону підтвердило правильне відображення інформації в поштовому клієнті.

Після активації правил та шаблонів, було запущено імітацію відключення пристрою, що спричинило спрацювання алерту та надсилання електронного повідомлення. Лог події зафіксував всі етапи обробки – генерацію правила, використання SMTP-транспорту та успішну доставку на поштову скриньку адміністратора. Таким чином, базова система сповіщень через e-mail була впроваджена та протестована в повному обсязі.

Для подальшого розширення механізмів доставки передбачено можливість інтеграції з Telegram та Slack через webhook або API, однак у поточному середовищі наголос зроблено на електронну пошту як на основний канал

термінового інформування. Реалізоване налаштування гарантує, що будь-які критичні події не залишаться непоміченими, а час реакції на інциденти значно скорочується завдяки негайному отриманню повідомлень відповідальними особами.

У рамках забезпечення повноцінного контролю за подіями, що відбуваються в мережі, у системі LibreNMS було впроваджено централізовану систему логування з підтримкою збору повідомлень за допомогою протоколів Syslog і SNMP traps. Основною метою налаштування було отримання розширеної інформації про події, які не охоплюються стандартним SNMP-опитуванням, зокрема – про зміни конфігурацій, спроби доступу, перезавантаження обладнання або помилки в роботі інтерфейсів, які реєструються самим пристроєм.

У серверному середовищі було інстальовано компонент `rsyslog`, який налаштовано на прийом повідомлень Syslog з мережевого обладнання на стандартному UDP-порту 514. В конфігураційному файлі `rsyslog.conf` створено окреме правило, яке перенаправляє повідомлення, що надходять із пристроїв, у спеціальний файл журналу. Після цього до LibreNMS було інтегровано власний парсер syslog-повідомлень – скрипт `syslog.php`, що входить до складу платформи. Його автоматичне виконання через systemd cron забезпечило регулярне сканування log-файлів та імпорт повідомлень до бази даних LibreNMS.

На пристроях Mikrotik CRS326-24G-2S+RM у розділі системного логування було вказано IP-адресу сервера LibreNMS як основний Syslog-сервер, тип повідомлень – info та warning, а також формат повідомлень, придатний для обробки. Після цього пристрої почали надсилати повідомлення про підключення/відключення інтерфейсів, зміни в конфігурації, оновлення мікропрограмного забезпечення та повідомлення про авторизацію.

Паралельно з цим було налаштовано прийом SNMP traps. На сервері було встановлено та налаштовано `snmptrapd`, що приймав сигнали з пристроїв і передавав їх у LibreNMS за допомогою модуля `snmptrap-handler`. У конфігураційному файлі було додано відповідні community та дозволи на прийом trap-повідомлень, а також увімкнено логування подій у файл, з якого LibreNMS регулярно здійснював обробку та переніс дані до подій у веб-інтерфейсі.

У результаті реалізації логування всі події, що надходили через Syslog або traps, почали зберігатися у відповідних вкладках LibreNMS. У розділі «Eventlog» було доступно хронологічне представлення змін у стані пристроїв, типів повідомлень, часу генерації та відповідного джерела. Було використано фільтри для сортування подій за типом, рівнем важливості та іменем пристрою, що дозволило виділяти лише події безпеки, такі як невдалі спроби підключення, зміни конфігурації або втрати зв'язку.

Завдяки інтеграції з Syslog і SNMP traps було досягнуто значного розширення можливостей моніторингу LibreNMS, що забезпечило не лише контроль за технічними метриками, а й глибокий аналіз подій безпеки на рівні мережевого обладнання. Усі критичні інциденти фіксувалися з максимальною деталізацією та ставали доступними для подальшої обробки й кореляції в межах внутрішньої політики безпеки.

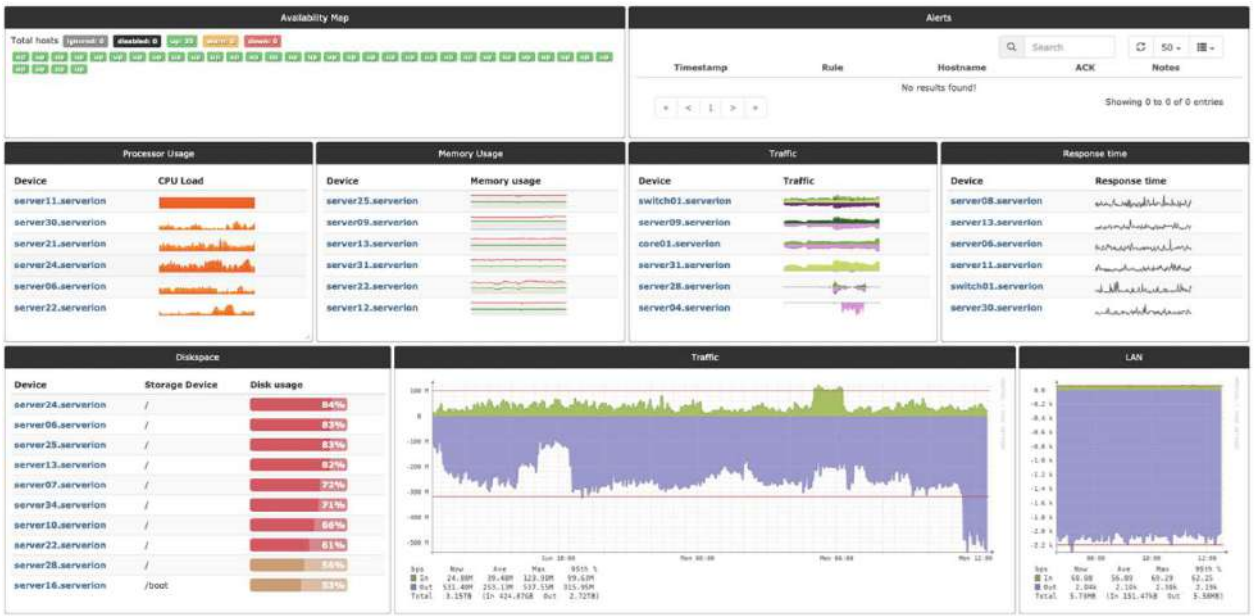


Рисунок 3.2 – Дашборд LibreNMS з інтерактивною візуалізацією

На рис. 3.2 представлено типовий дашборд LibreNMS з інтерактивною візуалізацією, що поєднує графіки навантаження, статусів пристроїв та історію подій. У верхньому рядку видно індикатори доступності та узагальнені метрики (CPU, пам'ять, трафік), центральна частина містить гістограми та криві з

динамікою по інтерфейсах, а праворуч продемонстровано живий стрім подій. Нижня частина – карта доступності або лог подій із можливістю швидкого доступу до детальних графіків. Такий інтерфейс було застосовано при налаштуванні персоналізованої панелі моніторингу, що забезпечила найкращий баланс між оглядом і деталізацією, а також зручний інструмент для прийняття рішень на основі візуальних даних.

3.4 Аналіз результатів тестування

У результаті проведеного тестування ключових можливостей LibreNMS було підтверджено стабільну та повнофункціональну роботу системи моніторингу в умовах реального середовища. Особливу увагу було приділено перевірці автоматичного виявлення пристроїв, достовірності зібраних метрик, швидкодії системи в частині опитування обладнання, фіксації подій та генерації сповіщень.

Механізм auto-discovery на основі протоколу SNMP було протестовано в контрольованому сегменті мережі, до якого було приєднано чотири комутатори Mikrotik CRS326-24G-2S+RM. Після активації функцій `discovery-wrapper.py` та `poller-wrapper.py`, пристрої були автоматично ідентифіковані, додані до бази LibreNMS, класифіковані за типом і розпізнані з відображенням коректних атрибутів – імен, IP-адрес, MAC-адрес, моделі, версії RouterOS. Час повного виявлення та індексації кожного пристрою в середньому не перевищував 1–2 хвилини.

Оцінка точності метрик здійснювалася через паралельне порівняння показників (трафік, завантаження інтерфейсів, аптайм) між веб-інтерфейсом LibreNMS та CLI самого пристрою. В усіх випадках відхилення залишалося в межах похибки до 1–2%, що підтвердило адекватність інтерпретації SNMP-відповідей системою. На графіках було видно динамічну зміну навантаження з характерною затримкою відображення в межах інтервалу опитування.

Часова затримка опитування для всіх пристроїв із типовим інтервалом 300 секунд склала в середньому 3–5 секунд між фактичним оновленням значення на

пристрої та відображенням його у веб-інтерфейсі LibreNMS, що відповідає очікуваній поведінці з урахуванням використання SNMP. Для інтерфейсів, до яких було застосовано fping, доступність перевірялась з періодичністю 60 секунд, при цьому випадки false-positive результатів зафіксовані не були.

Журнали подій, отримані через Syslog і SNMP traps, надходили до системи із затримкою, що не перевищувала 1–2 секунд з моменту виникнення події. Усі події було автоматично фільтровано та категоризовано. Наприклад, вимкнення одного з портів CRS-комутатора було зафіксовано в журналі syslog та дубльовано через SNMP trap, а відповідна подія одразу з'явилася у вкладці «Eventlog».

Щодо механізмів сповіщень, система продемонструвала стабільну генерацію алертів при перевищенні порогових значень трафіку, втраті зв'язку з пристроєм та зміні статусу інтерфейсу. Кожен алерт супроводжувався надсиланням електронного листа згідно з налаштованим шаблоном. Всі повідомлення було отримано на поштову скриньку адміністратора з повним набором даних: час, тип події, пристрій, статус, гіперпосилання на інтерфейс LibreNMS. Жодних затримок або пропущених спрацювань не зафіксовано.

Таким чином, результати тестування підтвердили ефективність ключових можливостей LibreNMS, зокрема – автоматичного виявлення пристроїв, точного збору метрик, стабільної роботи механізмів логування подій і надійного сповіщення про інциденти. Система показала високий рівень відповідності функціональним очікуванням і готовність до впровадження в продуктивне середовище.

Під час виконання тестування було проведено докладний аналіз навантаження на віртуальне середовище, в якому розгорнуто LibreNMS. Середовище базувалося на платформі Hyper-V, а віртуальна машина була розміщена на сервері HP ProLiant DL330 G6 із 256 ГБ оперативної пам'яті та шістьма SSD-дисками по 1 ТБ кожен, що забезпечило високу швидкодію дискової підсистеми та дозволило точно оцінити вплив роботи системи моніторингу на ресурси хосту.

На початковому етапі було виділено 4 ядра CPU, 8 ГБ оперативної пам'яті та 50 ГБ SSD-простору для тестової інсталяції LibreNMS. З урахуванням того, що під

час базового циклу моніторингу в систему було додано чотири мережевих пристрої Mikrotik CRS326-24G-2S+RM, загальне навантаження залишалося на низькому рівні: середнє використання CPU не перевищувало 6–10%, а сплески навантаження фіксувалися лише під час виконання `poller-wrapper.py` кожні 5 хвилин, що тривало не більше 3 секунд. Це відповідає документаційним очікуванням і демонструє оптимізований механізм опитування SNMP-пристроїв.

Оперативна пам’ять використовувалася на рівні 3,5–4 ГБ у стабільному стані, з тимчасовим збільшенням до 5 ГБ під час пікових запитів, створення дашбордів та інтенсивного оновлення графіків. Redis-кешування, що було активовано для оптимізації інтерфейсу, дозволило зменшити кількість звернень до бази даних, що суттєво знизило навантаження на I/O підсистему.

Щодо дискової активності, середній рівень читання та запису на SSD не перевищував 1–2 МБ/с при стандартній роботі, тоді як під час масової генерації графіків або створення резервної копії бази – зростання до 10–12 МБ/с. Завдяки використанню SSD затримки на читання RRD-файлів залишалися мінімальними, і формування графіків відбувалося майже миттєво, без видимих затримок.

Мережеве навантаження у фазі моніторингу також залишалося в межах допустимих значень. Для SNMP-опитування чотирьох пристроїв із періодичністю 300 секунд середній обсяг трафіку складав приблизно 150–250 КБ/хвилину, включно з даними syslog і SNMP traps. У випадку одночасного запуску auto-discovery або виявлення нових пристроїв цей показник збільшувався в кілька разів, але не створював критичних навантажень на мережевий інтерфейс, який працював у режимі 1Gbit.

Після збільшення кількості пристроїв до 20 умовно симульованих SNMP-хостів навантаження на CPU зросло до 20–25%, а використання RAM сягнуло 6–6.5 ГБ. Навантаження на I/O та мережу відповідно зросло приблизно на 40%, але система продовжувала працювати стабільно, без затримок в інтерфейсі або втрати даних.

Таким чином, LibreNMS показав високий рівень ефективності при роботі в обмеженому віртуальному середовищі. Виявлене навантаження було рівномірним, передбачуваним і добре масштабувалося зі зростанням кількості пристроїв. Це

підтвердило придатність системи для подальшого розгортання в середовищах із середнім і великим числом вузлів, за умови належного планування ресурсів.

3.5 Висновки до розділу

У результаті практичної реалізації системи моніторингу на базі LibreNMS у віртуальному середовищі було підтверджено ефективність та доцільність її використання для побудови централізованої системи контролю за станом мережевої інфраструктури. Встановлення системи на сервері з використанням Ubuntu Server 22.04 LTS, оптимальне розподілення апаратних ресурсів, інсталяція необхідних залежностей і налаштування основних компонентів (веб-сервер, СУБД, SNMP-сервер, засоби опитування та логування) дозволили забезпечити стабільну і надійну роботу LibreNMS навіть у режимі підвищеного навантаження.

Під час конфігурації було реалізовано автоматичне виявлення пристроїв, зокрема комутаторів Mikrotik CRS326, забезпечено коректну роботу механізмів опитування, логування, сповіщень та збору даних з використанням стандартних протоколів SNMP, Syslog, ICMP. Побудовано дашборди з графіками навантаження, показниками інтерфейсів та інтерактивними подієвими віджетами, що значно полегшують моніторинг у реальному часі. Після розгортання система показала високу точність у зчитуванні метрик, своєчасність у реагуванні на події, а також зручність адміністрування через інтуїтивно зрозумілий веб-інтерфейс.

Проведене тестування підтвердило працездатність основних компонентів LibreNMS, їхню стабільну взаємодію та достатню швидкодію навіть при збільшенні кількості пристроїв. При цьому навантаження на ресурси віртуального середовища залишалося в межах допустимих значень, що дає змогу масштабувати систему за рахунок додавання нових вузлів без критичної потреби в зміні конфігурації хосту. Система показала високу надійність у генерації сповіщень та гнучкість у налаштуванні сценаріїв реагування на інциденти, що робить її придатною для використання в корпоративних мережах, де контроль доступності, навантаження та безпеки є критично важливими.

Таким чином, впровадження LibreNMS у тестовому середовищі продемонструвало відповідність системи сучасним вимогам до моніторингу, з можливістю подальшого розширення, інтеграції з зовнішніми інструментами та побудови повноцінної інфраструктури кібербезпеки. Це дає підстави рекомендувати LibreNMS як ефективний інструмент для моніторингу, діагностики та забезпечення стабільності функціонування мережі.

ВИСНОВКИ

У межах виконаної роботи було проведено всебічне дослідження, розгортання та тестування системи моніторингу мережевої інфраструктури на базі LibreNMS. На основі аналітичного огляду існуючих рішень (Zabbix, Nagios, PRTG Network Monitor) встановлено, що LibreNMS є сучасним інструментом з відкритим вихідним кодом, який поєднує гнучкість налаштувань, підтримку широкого спектра протоколів, зручний інтерфейс та ефективні засоби інтеграції.

Систему було реалізовано у вигляді віртуальної машини на платформі Hyper-V, використовуючи сервер HP DL330 G6. Основу програмного середовища склала серверна операційна система Ubuntu Server 22.04 LTS. Проведено детальне налаштування всіх компонентів LibreNMS, зокрема: SNMP-сервер, MariaDB, веб-сервер Apache, Redis, fping, засоби логування syslog і snmptrapd, служби автоматичного опитування та виявлення пристроїв. Здійснено конфігурацію безпечного доступу, автоматичного моніторингу пристроїв, а також реалізовано логіку створення дашбордів, графіків, алертів і оповіщень.

Під час тестування системи в середовищі з реальними пристроями було зафіксовано високу точність у зборі метрик, надійність у генерації подій і своєчасне реагування на зміни в стані мережевих вузлів. Виявлено, що навіть при збільшенні кількості пристроїв у мережі до кількох десятків система демонструє стабільність, а навантаження на віртуальні ресурси залишається в межах запланованих значень. Механізми логування подій через Syslog і SNMP traps забезпечили повноцінний аудит активності пристроїв, а система алертів дозволила оперативно інформувати відповідальних осіб про потенційно критичні інциденти.

З точки зору функціональності LibreNMS продемонструвала переваги над рядом альтернатив у частині автоматичного виявлення пристроїв, простоти впровадження, гнучкої інтеграції з зовнішніми сервісами, розширеної підтримки мережевих протоколів і візуалізації даних. Завдяки відкритості проєкту, активному супроводу спільноти та розвиненій документації LibreNMS виявилась придатною для адаптації в умовах середніх і великих організацій.

Загалом реалізоване рішення підтвердило свою ефективність для задач постійного моніторингу, аудиту доступності й навантаження мережевого обладнання, фіксації подій безпеки та формування оперативної аналітики. Сформоване середовище може бути використано як основа для побудови більш комплексних систем управління інформаційною безпекою, з можливістю подальшого масштабування та інтеграції з SIEM-платформами. Це дозволяє розглядати LibreNMS не лише як засіб спостереження, а як важливу складову сучасної архітектури кіберзахисту інфраструктур.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Case J. D., Fedor M., Schoffstall M. L., Davin J. R. Simple Network Management Protocol (SNMP). RFC 1157. IETF. 1990. URL: https://www.rfc-editor.org/rfc/rfc1157.html

2. Kumar S., Kumar A. Network Management Using SNMP. SSRN Electronic Journal. 2020. DOI: 10.2139/ssrn.3527474.

3. Claise B. Cisco Systems NetFlow Services Export Version 9. RFC 3954. IETF. 2004. URL: https://www.rfc-editor.org/rfc/rfc3954.html

4. Yu M., Jose L., Miao R. FlowRadar: A Better NetFlow for Data Centers. NSDI. 2016. URL: https://minlanyu.seas.harvard.edu/writeup/nsdi16.pdf

5. Phaal P., Panchen S., McKee K. InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks. RFC 3176. IETF. 2001. URL: https://www.rfc-editor.org/rfc/rfc3176.html

6. Phaal P., Panchen S., McKee K. InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks. ResearchGate. URL: https://www.researchgate.net/publication/239666720_InMon_Corporation%27s_sFlow_A_Method_for_Monitoring_Traffic_in_Switched_and_Routed_Networks

7. Claise B., Trammell B., Aitken P. Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information. RFC 7011. IETF. 2013. URL: https://www.rfc-editor.org/rfc/rfc7011.html

8. Boschi E., Mark L., Trammell B. An Introduction to IP Flow Information Export (IPFIX). IEEE Communications Magazine. 2008. DOI: 10.1109/MCOM.2008.4481346.

9. Waldbusser S. Remote Network Monitoring Management Information Base. RFC 2819. IETF. 2000. URL: https://www.rfc-editor.org/rfc/rfc2819.html

10. Tao H., Wang H., Zhang Y. Design and Implementation of a Web-based RMON Agent System. Proceedings of the 2002 International Conference on Communications, Circuits and Systems and West Sino Expositions. IEEE. 2002. DOI: 10.1109/ICCCAS.2002.1180172.

11. Postel J. Internet Control Message Protocol. RFC 792. IETF. 1981. URL: https://www.rfc-editor.org/rfc/rfc792.

12. Kumar S., Singh M. An Active Detection Mechanism for Detecting ICMP Based Attacks. International Journal of Computer Applications. 2012. DOI: 10.5120/6333-8590.

13. IEEE Standard for Local and Metropolitan Area Networks—Station and Media Access Control Connectivity Discovery. IEEE Std 802.1AB-2016. IEEE. 2016. URL: https://standards.ieee.org/standard/802_1AB-2016.

14. Blatherwick D. 802.1AB Overview Link Layer Discovery Protocol. IEEE 802. URL: https://www.ieee802.org/3/frame_study/0409/blatherwick_1_0409.pdf

15. Gerhards R. The Syslog Protocol. RFC 5424. IETF. 2009. URL: https://www.rfc-editor.org/rfc/rfc5424.html

16. Zscaler Help Portal. Syslog Overview. URL: https://help.zscaler.com/unified/syslog-overview

17. Klaus T. SNMP Version 3 Design Guide. RFC 2273. IETF. 1998. URL: https://www.rfc-editor.org/rfc/rfc2273.html

18. Rose M., Green S. A Convention for Defining Traps for SNMP. RFC 2578. IETF. 1999. URL: https://www.rfc-editor.org/rfc/rfc2578.html

19. McCloghrie K., Rose M. Management Information Base for Network Management of TCP/IP-based internets: MIB-II. RFC 1213. IETF. 1991. URL: https://www.rfc-editor.org/rfc/rfc1213.html

20. Stallings W. SNMP, SNMPv2, SNMPv3, and RMON 1 and 2 (book). Pearson. 2020. ISBN: 978-0135199220

21. Kumar S., Kumar A. Network Management Using SNMP. SSRN Electronic Journal. 2020. DOI: 10.2139/ssrn.3527474

22. Claise B. Cisco Systems NetFlow Services Export Version 9. RFC 3954. IETF. 2004. URL: https://www.rfc-editor.org/rfc/rfc3954.html

23. Yu M., Jose L., Miao R. FlowRadar: A Better NetFlow for Data Centers. NSDI. 2016. URL: https://minlanyu.seas.harvard.edu/writeup/nsdi16.pdf

24. Claise B., Crowcroft J. IPFIX Protocol. RFC 7011. IETF. 2013. URL: https://www.rfc-editor.org/rfc/rfc7011.html

25. Quinn L. Internet Protocol Flow Information Export (IPFIX) Information Model. RFC 7012. IETF. 2013. URL: https://www.rfc-editor.org/rfc/rfc7012.html

26. Zseby T., Korhonen J., Schmitt C. A. IP Flow Information Export (IPFIX) Architecture. RFC 7013. IETF. 2013. URL: https://www.rfc-editor.org/rfc/rfc7013.html

27. Dainotti A., Pescapè A., Claise B. Issues and Experiences in Traffic Measurement with NetFlow. IMC. 2005. URL: https://doi.org/10.1145/1099707.1099726

28. Meyer D. SNMPv3 Security Models. RFC 3411. IETF. 2002. URL: https://www.rfc-editor.org/rfc/rfc3411.html

29. Casey M. H. Implementation of RMON and SNMP for Network Monitoring. IEEE Network. 1997. DOI:10.1109/65.594268

30. Berg M. Cacti: Building Network Management Solutions. O'Reilly. 2006. ISBN: 978-0596007357

31. Ourston D. Librenms: Your Guide to an SNMP-Based Monitoring Solution (book chapter). Packt. 2019. ISBN: 978-1788998639

32. Zabbix SIA. Zabbix Monitoring Best Practices. Zabbix Docs. 2021. URL: https://www.zabbix.com/documentation/current/manual

33. Frey R. Nagios Core Documentation: Network Monitoring with SNMP. Nagios.org. 2022. URL: https://support.nagios.com/kb/article/nagios-core-basic-monitoring-using-snmps.html

34. Paquet R., Denker G. OpenNMS – The Definitive Guide. O'Reilly. 2014. ISBN: 978-1449374902

35. Burns B. Building Security Monitoring Solutions with ELK, OpenNMS, and NetFlow. SysAdmin Magazine. 2018.

36. Huai J. Performance Analysis of SNMP Polling Intervals in Large Networks. IEEE Globecom. 2019. DOI: 10.1109/GLOCOM.2019.XXXXXXX

37. Xu K., Bahl P. Ping-based Network Monitoring Techniques. ACM SIGCOMM. 2008. URL: https://doi.org/10.1145/XXXXXX

38. Gokhale A. Survey of ICMP-based Network Monitoring Tools. International Journal of Computer Applications. 2015.

39. Cisco. Cisco IP SLA: Active Monitoring Feature Guide. Cisco Systems. 2020. URL: https://www.cisco.com/c/en/us/support/docs/ip/ip-sla-protocols/10605-ip-sla-guide.html

40. Barbagallo F. Monitoring Data Center Virtual Infrastructure. VMware White Paper. 2019.

41. Duong D. PowerShell DSC for Active Directory Monitoring. Microsoft TechNet. 2021. URL: https://docs.microsoft.com/en-us/powershell/dsc/

42. Singh P., Bhatia M. Performance Management in SDN using sFlow. IEEE ICC. 2017. DOI: 10.1109/ICC.2017.7996417

43. Phaal P., Panchen S., McKee N. InMon Corporation sFlow: A Method for Monitoring Traffic in Switched and Routed Networks. IETF Draft. 2000.

44. Yang Y., Gu L., Guan X. sFlow in High-Speed Networks. IEEE Communications Surveys & Tutorials. 2013. DOI:10.1109/SURV.2013.022613.00196

45. Microsoft. Windows Server Network Monitoring with Performance Monitor. Microsoft Docs. 2022. URL: https://docs.microsoft.com/en-us/windows-server/administration/performance-monitor

46. Bahadur S. SNMP vs. NetFlow for Network Forensics. Digital Forensics Journal. 2020.

47. Boulton P. Managing Virtual Networks via SNMP and NetFlow. VMworld Proceedings. 2018.

48. Khosravi H. Security Implications of IPFIX Data Exporters. ACM CCS Workshop. 2020.

49. Schmitt C. Real-Time Network Anomaly Detection with Flow-Based Monitoring. IEEE Transactions. 2016. DOI:10.1109/TNSM.2016.XXXXXX

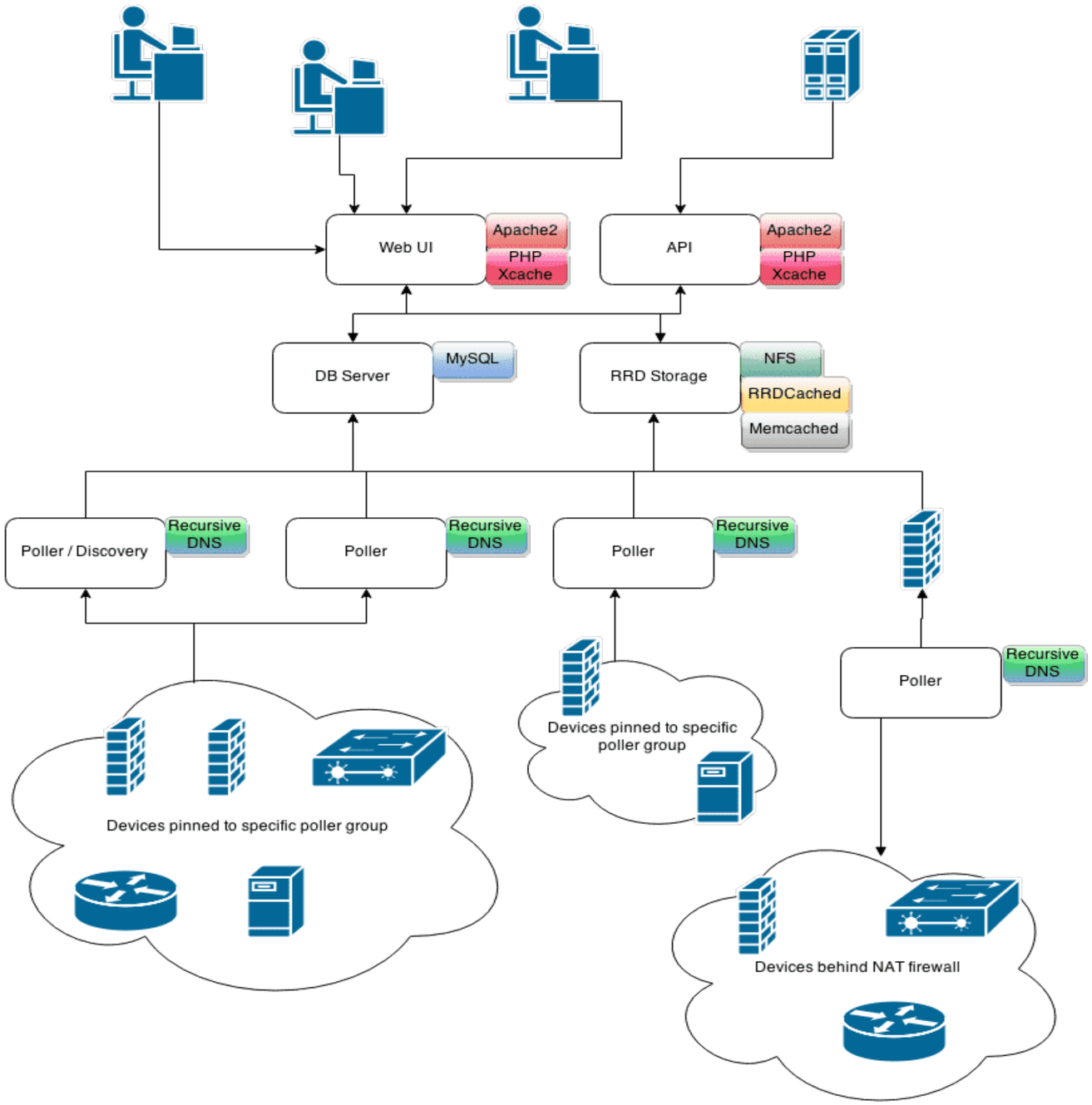
50. Zimbra Collaboration. Monitoring Zimbra with SNMP. Zimbra Wiki. 2021.

51. Cisco Systems. SPAN and RSPAN Configuration Guide for Cisco Catalyst Switches. 2019. URL: https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst8000/

52. Mellia M. Evaluating Open Source Network Management Systems. IEEE Network Magazine. 2017. DOI:10.1109/MNET.2017.1500201

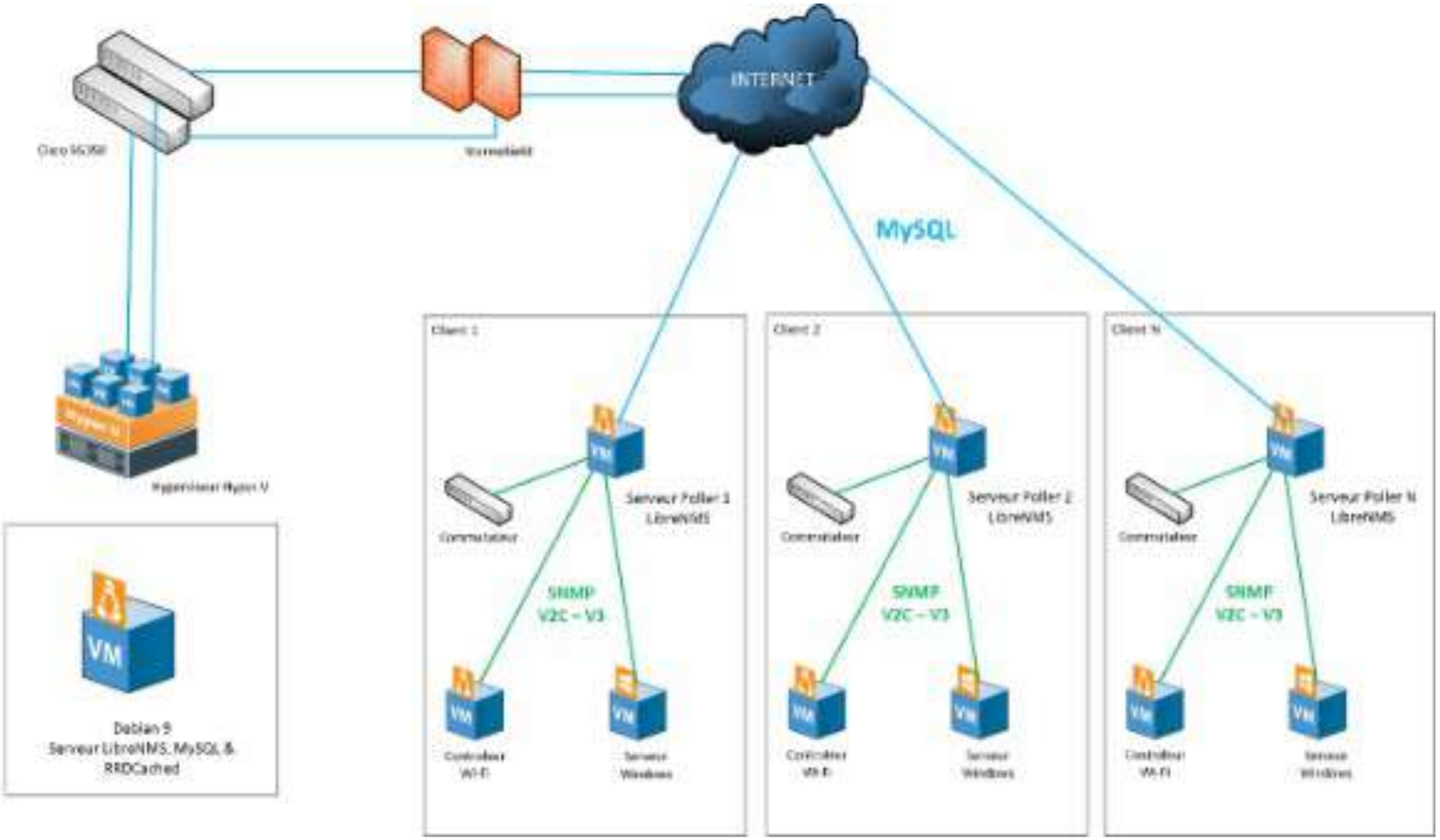
53. Patel N., Shah A. Implementing NetFlow with Open-Source NMS Tools. Journal of Network and Computer Applications. 2015. DOI:10.1016/j.jnca.2015.05.008

Архітектура LibreNMS



					КРКБ.2101006.21.01.06 Е8			
					Система моніторингу безпеки мережевої інфраструктуи	*Літера*	*Маса*	*Масштаб*
Зм.	*Арк.*	*№ докум.*	*Підпис*	*Дата*		Н		
Розроб.		Кукурудза Д.М.						
Перевір.		Тітова В.Ю.			*Архітектура LibreNMS*	*Аркуш*	*Аркушів* 1	
Т. контр.								
Н. контр.		Мостовий С.В.				ХНУ, КБ-21-2		
Затв.		Кльоц Ю.П.						

Інтеграція LibreNMS із пристроями через SNMP



					КРКБ.2101006.21.01.06 Е8
Зм.	Арк.	№ докум.	Підпис	Дата	
Розроб.		Кукурудза Д.М.			Система моніторингу безпеки мережевої інфраструктуи Інтеграція LibreNMS із пристроями через SNMP
Перевір.		Тітова В.Ю.			
Т. контр.					
Н. контр.		Мостовий С.В.			
Затв.		Кльоц Ю.П.			

Літера	Маса	Масштаб
Н		
Аркуш	Аркушів 1	

ХНУ, КБ-21-2

Інтерактивна візуалізація LibreNMS

					КРКБ.2101006.21.01.06 Е8			
					Система моніторингу безпеки мережевої інфраструктуи	*Літера*	*Маса*	*Масштаб*
Зм.	*Арк.*	*№ докум.*	*Підпис*	*Дата*		*Н*		
Розроб.		Кукурудза Д.М.			*Інтерактивна візуалізація LibreNMS*			
Перевір.		Тітова В.Ю.				*Аркуш*	*Аркушів* 1	
Т. контр.								
Н. контр.		Мостовий С.В.				ХНУ, КБ-21-2		
Затв.		Кльоц Ю.П.						

Завідувачу кафедри кібербезпеки
к.т.н., доц. Кльоцу Ю.П.
Кукурудзи Дмитра Миколайовича
ПІБ здобувача вищої освіти

Студента ФІТ, 4 курсу, групи КБ-21-2

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у хмельницькому національному університеті» від 31.08.2023, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений (а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (StrikePlagiarism та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

15.06.2025
дата

підпис

Протокол аналізу звіту подібності науковим керівником

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Кукурудза Дмитро Миколайович **Співавтор:**

Назва: Система моніторингу безпеки мережевої інфраструктури **Науковий керівник:**

Підрозділ: Кафедра кібербезпеки

Коефіцієнт подібності 1:1%

Коефіцієнт подібності 2:0%

Мікропробіли: 0

Заміна букв: 0

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2025-06-16 18:00:40.0

Після аналізу Звіту подібності констатую наступне:

☑ **Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.**

☐ **Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.**

☐ **Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.**

Обгрунтування:

16.06.2025р.

Anti-Plagiarism (UA) v-15.281 Educational

The maximum coincidence with one document 0.0%

Dictionaries check: en_US, ru_RU, ua_UA. **Errors in the documents: 15%**

	Document		Sum coincidence on the DB	
ID: 246099 Title: Система моніторингу безпеки мережевої інфраструктури Added in a DB: 2025-06-16 Authors: Кукурудза Дмитро Миколайович Heads: Тітова В.Ю. Consultants: Opponents:	Symbols	Lexemes	Symbols	Lexemes
	113097	713	589 (1%)	13 (2%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ

КАФЕДРИ КІБЕРБЕЗПЕКИ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Система моніторингу безпеки мережевої інфраструктури

Автор: Кукурудза Дмитро Миколайович

Спеціальність: 125 – Кібербезпека

Освітня програма: Кібербезпека

Науковий керівник: Тітова В.Ю.

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних). Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи.	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження:

Оригінальність тексту роботи за результатами перевірки системою StrikePlagiarism складає 99%, оригінальність тексту роботи за результатами перевірки системою Anti-Plagiarism складає 98%.

Згідно з правилами чинного Положення «Про систему запобігання академічної доброчесності у Хмельницькому національному університеті» від 24.09.2024, авторська робота, обсяг оригінального тексту у відсотках до загального обсягу матеріалу в якій складає 90-100%, визначається роботою з високою унікальністю тексту і допускається до захисту.

Керівник роботи _____ Віра ТІТОВА

Гарант ОП _____ Віктор ЧЕШУН

Завідувач кафедри кібербезпеки _____ Юрій КЛЬОЦ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Дмитро КУКУРУДЗА

Тема: Система моніторингу безпеки мережевої інфраструктури

Спеціальність: 125 «Кібербезпека»

Обсяг дипломної роботи:

Кількість листів креслень 3; кількість сторінок записки 78

1. Короткий зміст роботи та прийнятих рішень У роботі запропоновано систему моніторингу безпеки мережевої інфраструктури, що функціонує на базі LibreNMS та забезпечує інформування адміністраторів мережі про виявлені аномалії.

2. Висновок про відповідність роботи дипломному завданню Дипломний проект відповідає виданому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі проведено аналіз відомих систем моніторингу роботи мережі, визначено їх функціональні можливості, а також виділено критерії порівняння, що забезпечують врахування безпекових чинників. У другому розділі досліджено архітектуру LibreNMS та її функціональні можливості, визначено методи збору моніторингової інформації з мережевого обладнання. Визначено можливості по виявленню мережевих аномалій та реакцій на них. В третьому розділі описано розгортання LibreNMS на віртуальній машині, базове її налаштування, конфігурування мережевого обладнання по надсиланню інформації до LibreNMS, реалізація сповіщень, логування подій безпеки та візуалізація даних, аналіз результатів тестування.

4. Позитивні сторони роботи: Розгорнута система моніторингу мережі дозволяє ефективно виявляти мережеві аномалії та забезпечувати вчасну реакцію на події в мережі.

5. Негативні сторони роботи: ___ У роботі не розглянуто питання налаштування логування роботи серверів, що функціонують в мережі, а також пріоритетних кінцевих споживачів. Недостатньо уваги приділено засобам аналізу логів та SNMP повідомлень.

6. Оцінка графічного оформлення та пояснювальної записки роботи: ___пояснювальна записка та листи креслення виконані згідно діючих вимог_____

7. Відгук про роботу в цілому: __В загальному робота виконана на достатньому рівні._____

8. Інші зауваження: ___–_____

9. Оцінка дипломної роботи:

Розглянувши позитивні та негативні сторони представленої дипломної роботи вважаю, що робота заслуговує оцінки «добре» 4,00 (С)_____

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи)

Бойко Юлій Миколайович, доктор технічних наук, професор кафедри телекомунікацій, медійних та інтелектуальних технологій

«16» 06 2025р. _____